SAP CPQ Integration Guide
# Content

1 Integrations with SAP Solutions. ................................................................. 5  
1.1 System Requirements for SAP Integrations. .......................................... 5  
1.2 SAP ERP Integration. .................................................................................. 7  
  Troubleshooting Scenario. ............................................................................ 8  
  Defining Preselected Category. .................................................................... 9  
1.3 SAP S/4HANA Integration. .......................................................................... 10  
  Purpose. ....................................................................................................... 10  
  Preparation. ................................................................................................ 10  
  Configuration. ............................................................................................ 11  
  Manual Build Steps. .................................................................................. 75  
1.4 Variant Configuration Integration. ............................................................. 78  
  Introduction to the Variant Configuration Integration. ................................ 78  
  Integrate SAP CPQ with SAP Variant Configuration and Pricing. ............ 82  
  Define SAP General Attribute Mappings. ................................................... 86  
  Technical Overview. .................................................................................. 91  
  Pricing. ...................................................................................................... 98  
1.5 SAP Subscription Billing Integration. ......................................................... 106  
  Prerequisites. ............................................................................................. 106  
  Setup Adjustments on the SAP CPQ Side. .................................................. 107  
  Basic Integration Scenario. ......................................................................... 112  
  Create Subscription Product. .................................................................... 113  
  Set Values for Rate Plan Template Charges. .............................................. 115  
  Use Subscription Products on User Side. .................................................... 116  
  Quote Item Fields on Subscription Products. ............................................. 118  
1.6 SAP Cloud for Customer Integration. ......................................................... 119  
  SAP Cloud for Customer - SAP CPQ Integration. ..................................... 119  
  Legacy SAP Cloud for Customer - SAP CPQ Integration. ......................... 177  
1.7 SAP Commerce Cloud Integration. ........................................................... 185  
  Prerequisites for SAP Commerce Cloud Integration. ................................ 186  
  Integrate SAP CPQ with SAP Commerce Cloud. ..................................... 189  
  SAP Commerce Cloud Integration Scenarios. .......................................... 189  
1.8 SAP Commerce Cloud Integration for Quote 2.0. ...................................... 191  
  SAP Commerce Cloud User Scenarios. ...................................................... 191  
  Setting Up SAP Commerce Cloud Integration for Quote 2.0. ..................... 193  
  Field Mapping and Data Exchange. ............................................................ 199  
2 Subscription to Events. ................................................................................. 201
<table>
<thead>
<tr>
<th></th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>CRM Objects</td>
<td>276</td>
</tr>
<tr>
<td>12</td>
<td>Custom Fields Mapping</td>
<td>278</td>
</tr>
<tr>
<td>13</td>
<td>Customer Roles Mapping</td>
<td>280</td>
</tr>
<tr>
<td>14</td>
<td>Opportunity Status</td>
<td>282</td>
</tr>
<tr>
<td>15</td>
<td>CRM Item Mappings</td>
<td>283</td>
</tr>
<tr>
<td>16</td>
<td>Price Book Market Mappings</td>
<td>285</td>
</tr>
<tr>
<td>17</td>
<td>User Mappings</td>
<td>286</td>
</tr>
<tr>
<td>18</td>
<td>CRM Quote</td>
<td>287</td>
</tr>
<tr>
<td>19</td>
<td>CRM Administrator Account</td>
<td>288</td>
</tr>
<tr>
<td>20</td>
<td>Send Document To CRM</td>
<td>290</td>
</tr>
<tr>
<td>21</td>
<td>New Authentication method for COD integration</td>
<td>291</td>
</tr>
<tr>
<td>22</td>
<td>Netsuite Integration With SAP CPQ</td>
<td>292</td>
</tr>
</tbody>
</table>
1 Integrations with SAP Solutions

SAP CPQ can be integrated with a number of other SAP solutions.

System Requirements for SAP Integrations [page 5]
Before integrating SAP CPQ with another SAP application, please make sure that you meet all the necessary system requirements and own the required components.

SAP ERP Integration [page 7]
The SAP CPQ – SAP ERP integration keeps data synchronized between the two systems in real time.

SAP S/4HANA Integration [page 10]

Variant Configuration Integration [page 78]
The SAP CPQ – Variant Configuration integration unites front and back-office processes to simplify complex product configurations for sales representatives.

SAP Subscription Billing Integration [page 106]
Integrating SAP CPQ with SAP Subscription Billing allows you to conduct subscription-based business, which is gaining prominence in today’s economy.

SAP Cloud for Customer Integration [page 119]
The following pages describe both integrations and contain step by step instructions on how to integrate SAP Cloud for Customer and SAP CPQ.

SAP Commerce Cloud Integration [page 185]
The SAP CPQ – SAP Commerce Cloud integration increases the front office sales efficiency and enables B2B customers to directly interact with sales representatives via collaboration functionalities that both systems have.

SAP Commerce Cloud Integration for Quote 2.0 [page 191]
The integration between SAP CPQ and SAP Commerce Cloud allows SAP Commerce Cloud customers to negotiate on quote specifics and send a request for quotation to SAP CPQ in one click. They can add subscription and standard physical products and ask for discounts and change of quote terms by leaving comments on the quote and sending it to SAP CPQ.

1.1 System Requirements for SAP Integrations

Before integrating SAP CPQ with another SAP application, please make sure that you meet all the necessary system requirements and own the required components.

The provided information is valid for the current release of SAP CPQ.
<table>
<thead>
<tr>
<th>Application</th>
<th>System Requirements</th>
<th>Other Required Components</th>
</tr>
</thead>
</table>
| SAP ERP for simple products (without variant configuration) | SAP ERP 6.0 EHP 0 or higher                                                          | ● SAP Cloud Platform Integration  
● ERP C4 Integration Add-On CO-DERINT |
| SAP S/4HANA On-Premise                          | SAP S/4HANA version 1809 or higher                                                   | ● SAP Cloud Platform Integration  
● In case configurable products are in use: SAP Variant Configuration and Pricing |
| SAP ERP for variant configuration products       | SAP ERP 6.0 EHP 5 or higher (Unicode System). For detailed system requirements, see SAP Note 2711932.  
It is necessary to own an SAP back end solution with an appropriate database license that allows direct database access for data replication (3rd party DB, SAP Hana Platform). For further details and more information about interoperability and maintenance rules for Data Provisioning (DP) Agent, see SAP Note 2546811.  
Supported SAP ERP back-end databases per listed data provisioning adapters (DP Adapter) are listed in SAP Note 2546811.  
Cluster and pool table access in SAP ERP is only supported on UTF-16 Unicode databases. See SAP Note 2511048.  
● SAP Cloud Platform Integration  
● ERP C4 Integration Add-On CO-DERINT  
● SAP Variant Configuration and Pricing (comes with SAP CPQ, edition for variant configuration) |
| SAP Cloud for Customer                           | SAP Cloud for Customer version 1905 or higher  
SAML 2.0 Single Sign On needs to be established so that users are authenticated in SAP CPQ when navigating from SAP Cloud for Customer.  
● SAP Cloud Platform Integration  
● SAP ERP or SAP S/4HANA as a source system of product master and customer master (out of the box, SAP CPQ and SAP Cloud for Customer do not share the same product (material) master) |
| SAP Commerce                                    | SAP Commerce (on premise) version 1811 or higher, or SAP Commerce Cloud version 1811 or higher | ● SAP Cloud Platform Integration  
● In case configurable products are in use in SAP ERP or SAP S/4HANA: SAP Variant Configuration and Pricing. Works with SAP Commerce Cloud only. SAP Commerce on premise only supports simple products. |
1.2 SAP ERP Integration

The SAP CPQ – SAP ERP integration keeps data synchronized between the two systems in real time.

Key benefits of integrating SAP CPQ with SAP ERP are as follows:

- Products are synchronized directly from a remote SAP ERP system.
- Prices for simple products are sent from SAP ERP to SAP CPQ pricebooks over SAP Cloud Platform Integration.
- Product and pricing data is always up to date.
- Customers, together with all details, are sent from SAP ERP to SAP CPQ over SAP Cloud Platform Integration.
- Manual data maintenance is reduced.

**i Note**

To successfully integrate any system with SAP CPQ, the application parameter *Allowed origins for the CORS filter* needs to be configured in the way explained in Application Parameters.

**SAP ERP Integration on the SAP CPQ side**

To connect SAP ERP on the SAP CPQ side:

1. Go to | Setup | Providers | Providers | SAP |
   Available SAP providers display.
2. Click **SAP ERP**.
   A new page displays.
3. Set **Connect SAP ERP** to **TRUE**.
4. Click **Save**.
   A confirmation message displays.

Additionally, to be able to place orders to SAP ERP, you also need to define common SAP provider settings [page 178].

**i Note**

In | Setup | Product Catalog | Products/Products | products obtained via synchronization with SAP ERP and SAP Variant Configuration and Pricing have the value **Yes** in the column **Synced from Back Office**. The
Integration on the SAP ERP side

For step-by-step instructions on how to integrate SAP ERP with SAP CPQ on the SAP ERP side, visit SAP Best Practices for SAP CPQ integration with SAP ERP.

Troubleshooting Scenario [page 8]
This topic describes a typical troubleshooting scenario for the SAP ERP integration.

Defining Preselected Category [page 9]
A preselected category defines where products, which are synchronized from SAP ERP, will be placed in SAP CPQ catalog.

Related Information

System Requirements for SAP Integrations [page 5]

1.2.1 Troubleshooting Scenario

This topic describes a typical troubleshooting scenario for the SAP ERP integration.

Sales orders sent from SAP CPQ to SAP ERP may fail due to various reasons. In that case, the SAP CPQ quote will remain in the Order Confirmation Pending status and will therefore be read-only (for the regular Sales user type). This will prevent users belonging to the mentioned user type from editing the quote and optionally resending the order.

The recommended course of action is to create an additional user type which will be able to edit quotes in the Order Confirmation Pending status and change the status to the one where the Place Order to ERP action is available. Workflow needs to be adjusted accordingly.

Furthermore, there is a risk of creating double sales orders from the same quote (for example, a sales order failed because of a network issue). The user (belonging to the newly created user type that is able to edit the quote in the Order Confirmation Pending status) edits the quote and resends the sales order. In the meantime, network issues are fixed and the sales order that was created first arrives to the SAP ERP system. In that case, there will be two sales orders that came to SAP ERP from the same SAP CPQ quote.

In order to avoid this error, clients are recommended to check the logs in the SAP ERP system in order to determine the cause of the sales order failure, before they edit the SAP CPQ quote and optionally resend the sales order.
1.2.2 Defining Preselected Category

A preselected category defines where products, which are synchronized from SAP ERP, will be placed in SAP CPQ catalog.

SAP CPQ administrators can easily locate them and further modify them. Updating a product does not affect the product’s category, which means that the product category will be preselected only for the first time, and that only the product itself will be updated.

To define a preselected category:

2. On the Product Catalog tab, find the parameter Default category name for new products created via API.
3. In the dedicated field, enter a name for the preselected category (for example, Draft).
4. Save your changes.

Products will be added to the preselected category only when PreselectInCPQ=True in the <Categories> node in Simple Product Administration API.

If a preselected category is not defined, standard rules for product import will be followed. Products will be imported and assigned to the category that is defined in the simple product import API.

Example

The following is an abridged sample of the XML request payload.

```xml
<Products>
  <Product>
    <IsSAPProduct>True</IsSAPProduct>
    <Identifier>PartNumber</Identifier>
    <PartNumber><![CDATA[VC_DEMO_PLACEHOLDER]]></PartNumber>
    <UnitOfMeasure><![CDATA[q*3]]></UnitOfMeasure>
    <ProductType>Food</ProductType>
    <ProductSystemId><![CDATA[VC_DEMO_PLACEHOLDER]]></ProductSystemId>
    <Active>TRUE</Active>
    <ProductName>
      <![CDATA[VC_DEMO_PLACEHOLDER30]]>
    </ProductName>
    <Categories PreselectInCPQ="True">
      <![CDATA[Food]]>
    </Categories>
  </Product>
</Products>
```
1.3 SAP S/4HANA Integration

For detailed step-by-step instructions on how to set up the SAP S/4HANA integration with SAP CPQ, please refer to SAP Best Practices for SAP CPQ integration with SAP S/4HANA.

**i Note**

To successfully integrate any system with SAP CPQ, the application parameter *Allowed origins for the CORS filter* needs to be configured in the way explained in *Application Parameters*.

1.3.1 Purpose

The purpose of this document is to describe the procedure to configure the Basic Integration between SAP S/4HANA and SAP CPQ using SAP Cloud Platform Integration (CPI).

Network Security team takes the responsibility of preparing the network environment across different systems and its security aspects.

Important: If you have any queries or feedback on this document, please create a ticket for the CSN component SV-RDS-CFC.

1.3.2 Preparation

These are the preparatory steps that need to be followed:
1.3.2.1 Required Information

We recommend that the consultant performing the implementation has the following implementation and configuration knowledge:

- SAP S/4HANA
- SAP Cloud Platform Integration (CPI)
- SAP CPQ
- SAP CPS

1.3.2.2 Prerequisite

Before you set up SAP CPQ Integration with SAP S/4HANA, ensure that you have:

1. SAP CPQ instance
   Ensure that your SAP CPQ instance is in place before you start implementing SAP CPQ integration with SAP S/4HANA.

2. SAP S/4HANA System
   Ensure that your S/4HANA system is in place.

3. SAP Cloud Platform Integration tenant
   The SAP Cloud Platform Integration (CPI) Runtime URL is included in the SAP CPI provisioning e-mail. URL Example: https://<tenant>-iflmap.hana.ondemand.com.

4. SAP CPS instance (optional)
   Ensure that your SAP CPS instance is integrated to SAP S/4HANA system. And your configured products are replicated between SAP S/4HANA and SAP CPS.

1.3.3 Configuration

The following section describes all the settings required for this scope item. These can be divided into below groups:

Configuration in SAP S/4HANA [page 12]
This section describes how to configure SAP S/4HANA for the purpose of integration.

Configuration in SAP Cloud Platform Integration [page 47]
SAP provides prepackaged, generic integration content called integration flows (iFlows) for the integration of SAP CPQ with SAP S/4HANA using SAP Cloud Platform Integration (CPI).

Configuration in SAP CPQ [page 55]
This section describes how to configure SAP CPQ for the purpose of integration.

Initial Data Load [page 71]
This section describes how to execute the initial data load of material, customer master and Pricing Condition data from your SAP S/4HANA system to the SAP CPQ system.
1.3.3.1 Configuration in SAP S/4HANA

This section describes how to configure SAP S/4HANA for the purpose of integration.

1.3.3.1.1 Create User

1. Access the following transaction:

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>SU01</th>
</tr>
</thead>
</table>

2. On the User Maintenance: Initial screen, enter the following values:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>CPQS4HINTEG</td>
</tr>
</tbody>
</table>

3. Choose Create.
4. On the **Maintain User screen**, maintain the following values and click **Save**.

<table>
<thead>
<tr>
<th>Address Tab Page</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Name</td>
<td>&lt;user name&gt; example: CPQS4HINTEG</td>
</tr>
<tr>
<td>E-Mail</td>
<td>&lt;valid E-Mail Address to send Request for Quotations and Orders&gt;</td>
</tr>
<tr>
<td>Comm.Meth</td>
<td>Remote Mail</td>
</tr>
</tbody>
</table>

5. Logon data tab page

<table>
<thead>
<tr>
<th>User Type</th>
<th>Communication Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>&lt;password&gt;</td>
</tr>
</tbody>
</table>

**Note**

Ensure the user is assigned relevant authorizations to execute Idoc, web service messaging calls like external pricing and print preview.

6. Click **Save**.

### 1.3.3.1.2 Certificate Management in SAP S/4HANA

**Prerequisite**

- The customer network environment should be prepared. The SSL should be enabled on SAP S/4HANA ABAP Server. SAP S/4HANA Server/Client Certificates should be signed by the supported CA which is trusted by SAP Cloud Solution and SAP Cloud Managed Services.
- For preparing the network environment, network security team that takes care of the customer network is responsible.
- SSL needs to be enabled on SAP ABAP server by SAP Basis Consultant.

**Note**

Ensure all the certificates used are valid and are not expired.

For details refer to Note 510007 - Setting up SSL on Web Application Server ABAP.

For requesting signed certificates by SAP, go to [https://support.sap.com/tcs](https://support.sap.com/tcs).

- There are mainly two partners in the landscape, SAP side and Customer side.
- SAP Cloud Solution certificates are signed by SAP Passport CA or TC Trust Center.

#### Download ROOT CA Certificates

2. Navigate to Download > Root Certificates
3. Download the SAP Passport CA Certificates and Save.

- **Root Certificates**
  
  If you use web components (clients or servers) that do not already have these certificates, you can update these components by downloading the root certificates provided below.
  
  The different root certificates are used for different purposes, as described below. If you are not sure which one you need, you can import all of them.
  
  **Root CA Certificates of SAP Trust Center Services:**
  
  ![SAP Passport CA Certificate](https://example.com)
  
  - **SAP Passport CA Certificate**
    
    Users need this root certificate to verify SAP Passports. Therefore, you have to import this certificate in your web servers, for example, in front of your workplace or marketplace, to authenticate SAP Passport users.
  
  ![mySAP.com Test CA Certificate](https://example.com)
  
  - **mySAP.com Test CA Certificate**
    
    This root certificate has the same purpose as the SAP Passport CA certificate, but it is only used for testing and demo purposes.
  
  We protect the download of the root certificates by using the Secure Sockets Layer (SSL) Protocol. SSL will provide you with an encrypted and authenticated channel to the host tcs.mysap.com for downloading the root certificates.


Navigate using path **Monitor > Keystore** under **Manage Security**

5. Download the ROOT Certificates of CPI with following names:
   - **Baltimore CyberTrust Root**
   - **Verizon Public SureServer CA G14-SHA2**

**i Note**

In case of error encountered in downloading certificates, go to **Internet explorer > Tools > Internet options > content > Certificates > Trusted Root Certification authorities.** Export the relevant certificate from the list.
Maintain SSL Server Standard Trusted Cert List

1. In the SAP S/4HANA system access the following transaction:

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>STRUST</th>
</tr>
</thead>
</table>

2. Double-click **SSL server standard** PSE folder.
3. In the **Certificate** area, click the **Import Certificate** button to add the SAP Passport CA.
4. Add the imported certificate to the certificate list by clicking *Add to Certificate List.*

5. Click *Save.*
Maintain SSL System Client Cert and Trusted Cert List

1. Choose SSL client SSL Client(Standard) PSE folder.

2. From menu path choose Certificate > Import to import the SAP Passport CA certificate.

3. Switch from Display to Change mode.

4. From Menu Path choose Edit > Add Certificate to add the imported certificate to the chosen certification list.

5. Repeat previous 2 steps for Cloud Platform Integration Trust Root CA Certificates.

6. Click Save on the standard tool bar.

i Note

In case no supported CA signed client certificates for SAP S/4HANA is available, it could be downloaded from the SAP CPI Certificate store, and then imported into SAP Trust Manager for use.

Export SAP S/4HANA Client Certificate

1. Expand SSL client SSL Client (Standard) PSE folder.
   ○ Double-click target instance in the folder.
2. Select the field *Owner* in the *Own Certificate* section by double clicking.

3. Click the *Export* button.

4. Save the certificate into a file with format Base64 and named with `S4HANAClient.cer` for example.

**Note**

This exported client cert should be uploaded while configuring IFlow in CPI system.
Mapping Cloud Connector Certificate to Integration User in SAP S/4HANA

Purpose

Obtain the SAP CPI client certificate from your administrator. The certificate is available when the tenant is provisioned by SAP. You can also receive it on creating an incident in the component for your respective SAP Middleware (LOD-CPI/LOD-PI).

Procedure

1. Access the transaction using one of the following navigation options:

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>SM30</th>
</tr>
</thead>
</table>

2. In the Maintain Table Views screen, enter VUSREXTID in the Table/View field.

3. Click Maintain.

4. In the External ID type field, enter DN.

5. Click New Entries.
6. Next to the External ID field, click **Import**.

7. Import the Cloud Connector certificate obtained from your administrator.
   Example: `cloudconnectorclient.cer`

8. Enter the sequence number.
   Example: `000`

9. In the User field, enter the technical S/4HANA user you have created in the S/4HANA system. Refer to Section 3.1.1 Create User.
   For example: `CPQS4INTEG`

10. Activate the user by checking the Activated field.

11. Click **Save**.
1.3.3.1.3  Release Inbound Interfaces

Do the following steps to release inbound interfaces:

Activate Inbound SOAP for IDoc

1. Access the following transaction:

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>SICF</th>
</tr>
</thead>
</table>

2. In the Service Path field enter `/sap/bc/srt/IDoc`.

3. Click Execute.
4. Check if the service is active; if not then right-click the IDoc service and select *Activate Service*.

5. If it has been activated, it will be like below:

6. Otherwise you need to *Activate Service*.
Register Service for IDoc Inbound

1. Access the following transaction:

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRTIDOC</td>
<td></td>
</tr>
</tbody>
</table>

2. Select the Register Service radio button.

3. In the Service Attributes section, enter the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI SOAP Application</td>
<td>urn:sap-com:soap:runtime:application:idoc</td>
</tr>
<tr>
<td>Name of Web Service Definition</td>
<td>GENERIC</td>
</tr>
<tr>
<td>Call Address (ICF Path)</td>
<td>/sap/bc/srt/idoc</td>
</tr>
</tbody>
</table>

4. Click Execute.

5. Create a new request for this action and input the short description with meaningful value.

6. Click Ok.
1.3.3.1.4 Create Business System for SAP S/4HANA

This chapter describes the procedure to create a business system that represents the SAP S/4HANA system.

We can perform this task in following way.

1. Access the following transaction:

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>IDMIMG</th>
</tr>
</thead>
</table>

2. Navigate to BAdl, Determination of Local System Name from Define Technical Settings.
3. Execute the BAdl, Determination of Local System Name to create a local business system.

i Note
Check IMG Activity Documentation for more detail how to perform this task.
1.3.3.1.5 Define Logical System

This describes the procedure to create a logical system that represents the SAP CPQ system in SAP S/4HANA system. Logical systems are defined as cross-client systems.

1. Access the following transaction:

| Transaction Code | BD54 |

2. Click **New Entries**.

3. Create a logical system that identifies the on-demand system, as given in the table below:

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical System*</td>
<td>&lt;logical system name of SAP CPQ tenant&gt; Ex: CPQINTS4H</td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Description&gt; Ex: SAP CPQ integration with S/4HANA.</td>
</tr>
</tbody>
</table>

4. Click **Save**

5. Click **Ok** to confirm the pop-up for saving the details in the transport request.

1.3.3.1.6 Business Partner Replication from SAP S/4HANA to SAP CPQ

To replicate your business partners from SAP S/4HANA to SAP CPQ, perform the following steps:

Configure Data Replication Framework

1. Open transaction DRFIMG.

2. Navigate to **Define Custom Settings** for **Data Replication** ➔ **Define Technical Settings** ➔ **Define Technical Systems for Business Systems**.
3. Choose New Entries to add Business System for SAP CPQ. Add Business System name as logical system and Logical System name using input help. Choose Save.


5. Choose New Entries and add BO Type 986 (Business Partner including relationships) using input help and check field Sys. Filt. In case this entry exists, skip the step. Select the entry BO type and double click on
Define Bus. Systems, BOs, Communication Channel in the left dialog structure.

6. Select the entry BO type and double click on Define Bus. Systems, BOs, Communication Channel in the left dialog structure.

7. Choose New Entries to add the Communication Channel as 1 Replication via Services. In case the entry exists, skip this step.

8. Choose Save.
10. Select the IMG activity Define Replication Models.

11. Choose New Entries and define Replication Model Name, Description, and Log days as 50.

12. Choose Enter.
13. Select the entry created and double click on **Assign Outbound Implementation**.

14. Choose **New Entries**.
15. Select **Outbound Implementation** as Outbound Impl. for BP/REL via Services (986_3) using the input help and choose Enter.
16. Enter **Sequence** as 1.
17. Select the row added and double click on **Assign Target Systems for Repl. Model/Outb.Impl** in dialog structure.
18. Choose **New Entries** and add the Business system created above using the input help.
19. Select the Business system row and double click on the node **Assign Outbound Parameter** in the dialog structure.
20. Choose **New Entries** to add the Outbound parameter PACK_SIZE_BULK using the input help, check field **Mandatory** and enter outbound parameter value as 20.

<table>
<thead>
<tr>
<th>New Entries: Overview of Added Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assign Outbound Parameter</th>
<th>Parameter Description</th>
<th>Mandatory</th>
<th>Outbound Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACK_SIZE_BULK</td>
<td>Package Size for Bulk Messages</td>
<td>✔️</td>
<td>20</td>
</tr>
</tbody>
</table>

**Note**

Outbound Parameter value for PACK_SIZE_BULK can be set as per most efficient package size, which will differ from system to system.

21. Choose **Save**, details are saved in the transport request.

22. Click **Activate** to activate the replication model after selecting the folder Define Replication Model.

**Activate Function Modules**

1. Open transaction SPRO.

2. In IMG menu structure, navigate to | **Cross-Application Components** | **SAP Business Partner** | **Data Distribution** | **Activate Function Modules** |
Display IMG

Structure

- SAP HANA Cloud Platform Integration
  - Notification
  - SAP Jam Integration
  - European Monetary Union: Euro
  - Document Management
  - Classification System
  - Integriertes Produkt- und Prozess-Engineering (IPPE)
  - Engineering Change Management
  - Plant Data Collection
  - Time Sheet
  - Financial Conditions
  - General Application Functions
  - Bank Directory
  - Payment Cards
  - SAP Commercial Project Management
  - SAP Business Partner

- [ ] Activation Switch for Functions
  - Business Partner
- [ ] BADt: Additional Checks for Payment Cards
  - Business Partner Relationships
  - Business Partner Group Hierarchy
- [ ] External Data Transfer

Data Distribution

- [ ] Activate Function Modules
  - BADis for Data Replication
- Data Cleansing
3. Activate the function module MDG_BS_BP_OUTBOUND_DRF by checking the field Call as shown below.

4. Choose Save.

1.3.3.1.7 Material Replication from SAP S/4HANA to SAP CPQ using SOAP

To replicate material from SAP S/4HANA to SAP CPQ using SOAP, do the following steps:

Configure Data Replication Framework

1. Open transaction DRFIMG.
3. Choose **New Entries** to add **Business System** for SAP CPQ. If the Business system is already created, select **Business System** and **Logical System** name using input help. Choose **Save**.

4. Select the **Business System** and double click on **Define Business Systems**, BOs in the Dialog structure.

5. Choose **New Entries** and add BO Type 194 (Material) using input help and check field **Sys. Filt.** Select **Outp.Mode** as **Direct Output**. In case this entry exists, skip the step.

**Note**

If Output Mode is selected as Direct Output, idocs will be triggered immediately on Save. In this case, DRFOUT is not required to replicate changes to customer hierarchy. If Output Mode is selected as Pooled Output, idocs are sent using DRF change pointers and DRFOUT has to be executed or scheduled to create and send the Idoc.
6. Select the entry BO type and double click on **Define Bus. Systems, BOs, Communication Channel** in the left dialog structure.

7. Choose **New Entries** to add the **Communication Channel** as Replication via Services. In case the entry exists, skip this step.

8. Choose **Save**.
10. Select the IMG activity **Define Replication Models**.
11. Choose **New Entries** and define Replication Model Name and Description and Log days as 50.

12. Choose **Enter**.
13. Select the entry created and double click on **Assign Outbound Implementation**.

14. Choose **New Entries**.
15. Select **Outbound Implementation** as Outbound Impl. for Product via Services (194_3) using the input help and choose Enter.
16. Select the row added and double click on **Assign Target Systems for Repl. Model/Outb.Impl** in dialog structure.

17. Choose **New Entries** and add the Business system created above using the input help.
18. Select the Business system row and double click on the node Assign Outbound Parameter in the dialog structure.

![Assign Outbound Parameter in the dialog structure](image)

19. Choose New Entries to add the Outbound parameter PACK_SIZE_BULK using the input help. Enter outbound parameter value as 20 and click enter to check field Mandatory.

![Outbound parameter PACK_SIZE_BULK](image)

20. Choose Save.

21. Go back to Define Replication Model Screen and Select the row for Define Replication Model.

22. Click Activate to activate the replication model.

![Activate and Deactivate](image)

### 1.3.3.1.8 Configuration for Filter Values

Filter objects is used to remove the S/4HANA outbound processing code values which are not supported by the receiver system. During S/4HANA inbound processing these code values are added again.

**Filter Code Values**

> **Note**
> These below steps are optional and should be performed based on your requirement.

1. To setup Filter Objects for Code value, navigate to below path.

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>DRFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRFF</td>
<td></td>
</tr>
</tbody>
</table>
2. Execute Replication Model Filters Activity, it will open a pop up screen.

3. Enter the Login credential for your S/4HANA system.

4. Select the relevant Replication Model and choose display. It will open a different window to Display Filter Criteria.

5. Click on Edit and choose the Filter Criteria to Include Business Objects.

6. Select the BP role to be Include in your business Partner replication to SAP CPQ. For Example: FLCU01

7. Click on Show Segment filter and choose the Filter Object Business Partner Role-98601.

8. Choose Save.

### 1.3.3.1.9 Business Configuration Set

Business Configuration Set C4C_INTEGRATION_CS and C4C_INTEGRATION_CC contains additional customizing setting for SAP CPQ and SAP S/4HANA integration. BC set C4C_INTEGRATION_CC contains client-independent customizing entries for the ALE message types. Activate this BC set in the SAP S/4HANA client you use for cross-client customizing.

#### Procedure to activate Client Specific BC SET

1. Open transaction SCPR3

![Business Configuration Sets: Display](image)

3. In the dialog box enter BC Set name C4C_INTEGRATION_CS and choose Execute.

![New BC Set Selection](image)

4. Double click on the BC Set ID and choose Continue in the BC Set Consistency Check pop-up window.

5. Choose Copy BC Set

6. In the following dialog box, enter the new BC Set name and short text.
   Example: Z_CPQ_INTEGRATION_CS and SET CPQ Integration Customizing - client specific.

7. Choose Local Object or select a transport request if required.

8. Choose Define Favorites

9. Choose Change mode and choose Continue in the BC Set Consistency Check dialog box.

10. Right click on the node Maintenance View Variant for C4C CO Scenario in the structure and choose Remove Cust. Object from BC Set.
11. Right click on the node *Business Transaction Events* in the structure and choose *Remove IMG Activity from BC Set*.

12. Choose *Save*.


14. In the field BC Set, input the BC Set Name `Z_CPQ_INTEGRATION_CS`.

15. Choose *Activate BC Set (F7)* on the application tool bar.

16. Save it in a customizing request if there is a dialog box.
17. In the next pop-up, choose *Continue Activation* with default value.

**Procedure to Activate Cross Client BC SET**

1. Open transaction SCPR3

**Note**

Ignore if there are any warning messages during the activation of the BC set.
2. Choose **New BC Set Selection** ➔ **Select by Texts**

![Business Configuration Sets: Display](image)

3. In the dialog box enter BC Set name C4C_INTEGRATION_CS and choose **Execute**.

![New BC Set Selection](image)

4. Double click on the BC Set ID and choose **Continue** in the **BC Set Consistency Check** pop-up window.
5. Choose **Copy BC Set**
6. In the following dialog box, enter the new **BC Set name** and **short text**.
   Example: Z_CPQ_INTEGRATION_CS and SET CPQ Integration Customizing - cross client.
7. Choose **Local Object** or select a transport request if required.
8. Open transaction SCPR20
9. In the field BC Set, input the BC Set Name Z_CPQ_INTEGRATION_CS.
10. Choose *Activate BC Set (F7)* on the application tool bar, and create a new request for this.

![Prompt for Workbench request](image1)

**Business Configuration Sets: Activation Logs**

![Activation Log](image2)

**Note**
Ignore if there are any warning messages during the activation of the BC set.

### 1.3.3.1.10 Automatically Generate Integration Settings for Data Exchange

1. To Automatically Generate Connectivity settings for Data Exchange navigate using one of the following paths.

   **Transaction code**  
   RCOD_CONNECTIVITY

2. **Generate Integration Settings: Step 1**: Select *SAP Cloud Platform Integration* as the middleware and click *Next*.

   ![Generate Integration Settings: Step 1 of 5(6) - Middleware Selection](image3)

3. **Generate Integration Settings: Step 2**: Select the scope of Integration exactly as scoped in the SAP CPQ Solution and choose *Next*. 
4. **Generate Integration Settings: Step 3**: Maintain the following and choose **Next**.
5. **Generate Integration Settings: Step 4:** Enter the connection information about the CPI system to generate the RFC destinations. Choose Next.

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime URL</td>
<td>Worker node URL of the SAP Cloud Platform Integration tenant (without HTTPS ://).</td>
</tr>
<tr>
<td>Proxy details (Optional)</td>
<td>Host name, service name, and the access details of the forward or reverse proxy?</td>
</tr>
<tr>
<td>Authentication (basic or certificate)</td>
<td>Host name, service name, and the access details of the forward or reverse proxy?</td>
</tr>
</tbody>
</table>

6. **In Generate Integration Settings: Step 5 of 6-Consumer Proxy Settings** screen, enter the URL Access path for Business Partner and Product Replication. Enter the CPI system details in Connection Parameters and choose Certificate based authentic * /ation of CPI system.
### Note
You can take the URL Access Path from relevant IFlow of your CPI Web UI and for that service path add /cxf and maintain against the respective service.

<table>
<thead>
<tr>
<th>Generate Integration Settings: Step 5 of 6 - Consumer Proxy Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL Access Path</strong></td>
</tr>
<tr>
<td>Business Partner: /cxf/S4/CPQ/BusinessPartner/SUTEBulkReplicateRequest_Out</td>
</tr>
<tr>
<td>Business Partner Relationship: /</td>
</tr>
<tr>
<td>Product Replication: /cxf/S4/CPQ/ProductMDM BulkReplicateRequest</td>
</tr>
<tr>
<td>Attachment Replication: /</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Connection Parameters</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical Port</strong></td>
</tr>
<tr>
<td>CPQ_ILOGPORT</td>
</tr>
<tr>
<td><strong>Computer Name</strong></td>
</tr>
<tr>
<td>ifmap.hciab.int.sap.eu2.hana.ondemand.com</td>
</tr>
<tr>
<td><strong>URL Port</strong></td>
</tr>
<tr>
<td>443</td>
</tr>
<tr>
<td><strong>URL Protocol</strong></td>
</tr>
<tr>
<td>Https</td>
</tr>
<tr>
<td><strong>Proxy Host</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Proxy Port</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Proxy User</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Proxy Password</strong></td>
</tr>
<tr>
<td>****************************************************************</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Logon Procedure</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>* SSL Client Certificate</td>
</tr>
<tr>
<td><strong>Basic Authentication</strong></td>
</tr>
</tbody>
</table>

| SSL Client Certificate                                      |
| DEFAULT SSL Client (Standard)                               |

7. Choose **Next**.
8. In the **Summary** screen, select the **Generate** button to generate the configuration entities.

9. In the confirm pop-up screen choose **Yes**.
10. In the log you can check which entities were created.

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>29.03.2019</td>
<td>Report was executed for logical system CPQINTS4H</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Distribution CPQINTS4H.OPP_CONF of type G has been created</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Port CPQINTS4H.OPP_CONF of type G has been created</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Port CPQINTS4H.OPP_CONF of type G has been created</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Port CPQINTS4H.OPP_CONF of type G has been created</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Partner profile CPQINTS4H created</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Inbound message CPQINTS4H.OPP_CONF has been inserted into partner profile CPQINTS4H</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Message CPQINTS4H.OPP_CONF has been inserted into partner profile CPQINTS4H</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Message CPQINTS4H.OPP_CONF has been inserted into partner profile CPQINTS4H</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Message CPQINTS4H.OPP_CONF has been inserted into partner profile CPQINTS4H</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Message CPQINTS4H.OPP_CONF has been inserted into partner profile CPQINTS4H</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Message CPQINTS4H.OPP_CONF has been inserted into partner profile CPQINTS4H</td>
</tr>
<tr>
<td>29.03.2019</td>
<td>Message CPQINTS4H.OPP_CONF has been inserted into partner profile CPQINTS4H</td>
</tr>
</tbody>
</table>

1.3.3.11 Pricing Condition Filter

This describes the procedure to create an IDoc filter in SAP S/4HANA system. To allow only PPRO Pricing Condition data to flow to SAP CPQ system.

1. Access the following transaction:

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>BD64</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BD64</td>
</tr>
</tbody>
</table>

2. The Distribution Model for CPQ will be created by connectivity report, Search and Expand the Distribution Model **SAP CPQ Integration with S/4HANA**.

3. Expand the **COND_A** message type and double click on **No filter set**.
4. In a Change Filter popup, click on **Create Filter Group** button.
5. Expand the Filter Group and select the field **Condition Type**.
6. Click on Add button to add a value to filter field PPR0.

7. Click Ok button in popup windows and Save the Distribution Model.

1.3.3.2 Configuration in SAP Cloud Platform Integration

SAP provides prepackaged, generic integration content called integration flows (iFlows) for the integration of SAP CPQ with SAP S/4HANA using SAP Cloud Platform Integration (CPI).

To be able to import and deploy iFlows, you need the AuthGroup.IntegrationDeveloper role assigned in your tenant.

Upload SAP CPQ SSL Certificates in SAP Cloud Platform Integration.

1.3.3.2.1 How to get the SAP CPI client certificate

In the SAP CPI provisioning e-mail follow the link under Certificate Information. In the resulting screen choose X509 Certificate with option Binary CER. Download it to your local machine.
1.3.3.2.2 View prepackaged iFlows using SAP CPI Web UI

1. Access the web UI URL from the provisioning e-mail. It should be in the format: https://<hcitanent>.hana.ondemand.com/itspaces.
2. View all pre-packaged integration flows delivered by SAP in the Discover tab.
3. Click on the package SAP CPQ Integration with SAP S/4HANA.
4. You can see all the artifacts for this Integration Package.

1.3.3.2.3 Creating User Credentials in CPI System

1. Connect to the tenant management node of the SAP CPI system with the URL http://<tenant management node URL>/itspaces.
2. Go to the Monitor tab, then Click on Security Material under Manage Security.
3. Click Add and select User Credentials.

4. Enter the User detail for SAP CPQ.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>COM_USER_CPQ</td>
</tr>
<tr>
<td>User</td>
<td>COM_USER_CPQ#{DOMAIN of CPQ System}</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password maintained for Communication User in SAP CPQ.</td>
</tr>
</tbody>
</table>
5. Click **Deploy**.
6. Click **Add** and select **User Credentials**.
7. Enter details for user of SAP S/4HANA.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>CPQS4HINTEG</td>
</tr>
<tr>
<td>User</td>
<td>CPQS4HINTEG</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password of User CPQS4HINTEG.</td>
</tr>
</tbody>
</table>

8. Click **Deploy**.

**Note**

In case you have SAP CPS Integrated with SAP S/4HANA, then

1. Click **Add** and select **OAuth2 Credentials**.
2. Enter details for OAuth2 Credentials of SAP CPS.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>CPSSERVICE</td>
</tr>
</tbody>
</table>
1.3.3.2.4 Configure and Deploy the iFlows Using Web UI

If you have re-deployed key store (system.jks) in your SAP CPI tenant, you need to re-deploy all the iFlows to avoid the caching issues or re-start them to avoid caching related issues.

1. Connect to the tenant management node of the SAP CPI system with the URL `http://<tenant management node URL>/itspaces`.
2. Under the Discover tab, Click on the package SAP CPQ Integration with SAP S/4HANA.
3. Choose Copy on the top-right corner of the package overview page.
4. If the Integration package were being created for the first time, then you would see the message 'Integration Package Created'. If not, you will see the below dialog box asking to either create a new copy of the package or to overwrite the existing integration package content. Choose Overwrite.

---

Invalid Package Could not copy package, as it already exists in customer workspace; check the list of duplicated resources

Overwrite  Create copy  Close
5. Select the **Design** mode to configure the iFlows

6. Select the Integration Package copied.

7. All the iFlows will be shown/ listed under the **ARTIFACTS** tab of the package.

8. Choose the iFlows mentioned in the table below (one after the other) by choosing **Actions ➤ Configure**.

9. Configure the **sender system** (in this example S/4HANA) and **receiver system** (in this example CPQ) details as explained below:

   1. Choose **Sender Tab**: Choose **Authentication Method** as **Client Certificate**.
   3. Choose **Receiver Tab**: Enter the **Address** as CPQ API URL. Check section **How to find CPQ API URL** in appendix.
   4. Choose the **Authentication as Basic**.
   5. Enter the Communication User Credential in **Credential Name**.
   6. Click **Save** and Choose **Deploy**.
Note
Repeat the same procedure (From 9a. to 9f. as mentioned above) for all the iFlows that have the same sender system as S/4HANA and Receiver System as CPQ.

10. Choose iFlow to configure the sender system (in this example CPQ) and receiver system (in this example S/4HANA) details. :

   1. Choose Sender Tab: Choose Authorization as User Role.
   2. Add Address as /CPQ/S4/Quote2Order.
   3. Choose Receiver Tab: Select Receiver as S/4HANA.
   4. Enter the host name as Cloud Connector URL, check section How to setup Cloud Connector or Contact your system administrator for these details.
   5. Enter Client as the SAP S/4HANA system client. Proxy Type as On-Premise.
   6. Choose the Authentication as Basic.
   7. Enter the S/4HANA User Credential in Credential Name.
   8. Choose More Tab: Enter <SID>CLNT<Client> in Recipient Partner Number. For Example: QKXCLNT260.
   9. Enter SAP:<SID> in Recipient Port. For Example: SAPQKX.
   10. Enter Logical System you have created in S/4HANA in Sender Partner Number and Sender Port. For Example: CPQINTS4H.
   11. Choose Save and Choose Deploy.

iFlows to be configured

<table>
<thead>
<tr>
<th>Name of the iFlow</th>
<th>Description of the iFlow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Sales Quote Follow Up Document in SAP Business Suite</td>
<td>Creation of Sales Quote Follow Up Document in SAP Business Suite</td>
</tr>
<tr>
<td>Replicate Business Partner from SAP Business Suite</td>
<td>Business Partner Replication from SAP Business Suite</td>
</tr>
<tr>
<td>Replicate Material from SAP Business Suite</td>
<td>Material Replication from SAP Business Suite</td>
</tr>
<tr>
<td>Replicate Price Condition from SAP Business Suite</td>
<td>Price Condition Replication from SAP Business Suite</td>
</tr>
<tr>
<td>Replicate Sales Order Confirmation from SAP Business Suite</td>
<td>Creation of Sales Quote Follow Up Document in SAP Business Suite</td>
</tr>
</tbody>
</table>

i Note
In case you have SAP CPS Integrated with SAP S/4HANA, And Product replication done from SAP S/4HANA to SAP CPS. You can configure the Iflow “Create Sales Quote Follow Up Document in SAP Business Suite” for receiver CPS as well to check configurable product characteristics in SAP CPS follow the below steps.

   1. Choose Receiver Tab: Select Receiver as CPS.
   2. Enter the host name as SAP CPS URL or Contact your system administrator for these details.
   3. Enter Proxy Type as Internet and Select Method as GET.
   4. Choose the Authentication as OAuth2 Client Credentials.
   5. Enter the CPS OAuth2 Credential in Credential Name.
1.3.3.2.5 Value Mapping for CPQ integration with SAP S/4HANA

1. Connect to the tenant management node of the CPI system with the url: http://<tenant management node>/itspaces.
2. Go to Design view.
3. Select the Integration Package in your workspace.
4. All the iFlows will be shown/ listed under the Artifacts section of the page.
5. Select the iFlows with name Value Mapping for SAP CPQ integration with SAP S4HANA and choose Actions button on the right and select Configure.
   All the Mapping Values will be shown/ listed.

6. Map all the Value mapping fields as per below sections.
7. Choose Save and then Deploy.

**Product Type**

1. Mention the Material Group in SAP S/4HANA system in left side.
2. Enter the respective Product Type in CPQ system in right side. Maintain the Product Type created in 3.3.3.1 Maintain Product Type section.

**Product Category**

1. Mention the Product Hierarchy in SAP S/4HANA system in left side.
2. Enter the respective Product Category in CPQ system in right side. Maintain the Product Category created in 3.3.3.2 Maintain Product Category section.

**Role Code**

1. Mention the Business partner role code in SAP S/4HANA system in left side.
2. Enter the respective Business partner role code in CPQ system in right side. Maintain the Business partner roles available in CPQ system Customer Role Default under Customer/ Customer Role.

**Country**

2. Enter the respective Country Code in CPQ system in right side. Maintain the Countries Abrev3 available in CPQ system under General as Countries.

**Region**

1. Mention the Region Code in SAP S/4HANA system on left side. You can check Region Code in S/4HANA in T005S Table.
2. Enter the respective Region Code in CPQ system in right side. Maintain the States Abrev2 available in CPQ system under General as States.
Pricing Condition

1. Mention the Pricing Condition Code in SAP S/4HANA system on left side.
2. Enter the respective Pricing Description to be used in CPQ system in right side.

1.3.3.3 Configuration in SAP CPQ

This section describes how to configure SAP CPQ for the purpose of integration.

1.3.3.3.1 Define Provider settings

1. Login to your SAP CPQ system using an Administrator User.
2. Click on **SETUP** in menu bar and choose **Setup**.

3. Select **Providers** in left navigation bar and then select **SAP**.


5. In Common Settings tab, switch on **Enable Integration**.
6. Select **BasicAuth** in Authentication Mode dropdown.
7. Maintain Integration **Username** and **password** as the username and password of Cloud Platform Integration

![Integration Details](image)

8. Under **General Setting**, Switch on **Send customers along with quote payload**.
9. Enter the Cloud Platform URL in **REST API base URL address** and a postfix `/http` to connect to http adapter.

![General Settings](image)

10. Click **Save**.
11. Navigate to **S/4 ERP Settings** tab, maintain the endpoint as `/CPQ/S4/Quote2Order`.
12. Click **Save**.
13. Select **SAP ERP** in **Available SAP Providers**.
14. Select **Connect SAP ERP** as **TRUE**.
15. Click **Save**.
1.3.3.3.2  Create CPQ Communication User

A Communication user need to be created in SAP CPQ System, which will be used in Cloud Platform Integration for basic authentication to CPQ.

1. Login to your SAP CPQ system using an Administrator User.
2. Click on SETUP in menu bar and choose Setup.
3. In left navigation bar, select Users under Users.
4. Click on Add New User Button.
5. Enter the First Name, Last Name and Email Address for Communication User.
6. Enter Username as COM_USER_CPQ and maintain the Initial Password.
7. Choose User Type as Sales.
8. Select Administrator.
9. Select Company as CallidusCloud (CALD).
10. Click Save.

**Note**
Login to CPQ system using the above user and reset the password and use same password to be used in CPI at the time of Creating User Credentials.

1.3.3.3.3  Product Replication Setup

Do the following steps to replicate the product setup:

**Maintain Product Types**

**Purpose**
Follow below procedure to create Product Type in SAP CPQ. This Product Type is mapped to S/4HANA Material Group in SAP CPI middleware.

**Procedure**

1. Login to your SAP CPQ system using an Administrator User.
2. Click on SETUP in menu bar and choose Setup.
3. In left navigation bar, select Product Types under Product Catalog.
4. Click on Add New Button.
5. Maintain the Product Type Name and Product Type Rank.

**Example**

<table>
<thead>
<tr>
<th>Product Type</th>
<th>Product Type Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metal Processing</td>
<td>10</td>
</tr>
</tbody>
</table>
Product Type | Product Type Rank
---|---
Electronics | 11

**Product Types Administration » New**

<table>
<thead>
<tr>
<th>*Product Type Name</th>
<th>Metal Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Type System Id</td>
<td>Metal_Processing_1_cpq</td>
</tr>
<tr>
<td>Product Type Icon</td>
<td></td>
</tr>
<tr>
<td>*Product Type Rank</td>
<td>10</td>
</tr>
</tbody>
</table>

6. Click **Save**.

**Maintain Product Categories**

**Purpose**
In below steps, product catalog is created, which will help user to differentiate the product based on their category. If product doesn’t have any Product Category, it is considered as DRAFT. The Product Category is mapped with S/4HANA Product Hierarchy in CPI middleware.

**Procedure**
1. In left navigation bar, select **Categories** under **Product Catalog**.
2. Click on **Add New** Button.
3. Maintain the Product Category by giving **Category Name**, **Category Description**, **Rank** and check the **Active** checkbox.

**Example**

<table>
<thead>
<tr>
<th>Category Name</th>
<th>Category Description</th>
<th>Rank</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machines</td>
<td>Machines</td>
<td>10</td>
<td>X</td>
</tr>
<tr>
<td>Vehicles</td>
<td>Vehicles</td>
<td>11</td>
<td>X</td>
</tr>
</tbody>
</table>
4. Click Save.

**Maintain Unit of Measurement**

1. In left navigation bar, select *Unit of Measurement* under *Product Catalog*.
2. Click on *Add New* Button.
3. Enter the Description as *Measurement Name* and Code as *Unit of Measurement*. 
4. Click on Add New Button.

5. Click Save.

### 1.3.3.4 Customer Replication Setup

Do the following steps to replicate the customer setup:

#### Maintain Customer Role

1. Login to your SAP CPQ system using an Administrator User.
2. Click on SETUP in menu bar and choose Setup.
3. In left navigation bar, select Customer Role Defaults under Customers/ Customer Roles.
4. Click on Add New Button.
5. Select the User Type and Customer Role.
### Example

<table>
<thead>
<tr>
<th>User Type</th>
<th>Customer Role</th>
<th>User Info</th>
<th>User’s Company Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partner</td>
<td>Bill To</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Sales</td>
<td>Ship To</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

---

### Customer Role Defaults » Edit

- **User Type Name:** Sales
- **Customer Role:** Ship To

**Default (Pre-populate) with User Info:**
- **Default (Pre-populate) with User’s Company Info:**

[Save]

---

6. Click **Save**.

---

### Note

Customer Role Defaults settings has a significance in maintaining the Bill To and Ship To in the Customer Info tab of the CPQ quote hence maintain this setting as the User Type used for creating the quote.

---

## Maintain Customer Field Permissions

### Purpose

Business can define which fields of customer to be made mandatory and which should be read only. Below steps deals with minimum fields needed for customer replication from SAP S/4HANA system.

### Procedure

1. In left navigation bar, select **Customer Field Permissions** under **Customers/Customer Roles**.
2. Go to **Ship To** tab, click on the **permission** field and Select **Required** for below fields:

<table>
<thead>
<tr>
<th>Fields</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td></td>
</tr>
<tr>
<td>Company</td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td></td>
</tr>
</tbody>
</table>
**Note**

Maintain the field permissions as per your business needs, setting field permission as Required will make the field mandatory in the customer master, hence make sure that all the mandatory field values are available while business partner master data replication from S/4HANA system.

### Customer Fields Permissions

<table>
<thead>
<tr>
<th>Fields</th>
<th>Permissions</th>
<th>Advanced Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Last Name</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>Customer Id</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>Company</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Address (1)</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>Address (2)</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>State/Province</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>ZIP/Postal Code</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Territory</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>Fax</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>CRM Account Id</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>CRM Contact Id</td>
<td>Editable</td>
<td></td>
</tr>
<tr>
<td>Province</td>
<td>Editable</td>
<td></td>
</tr>
</tbody>
</table>

3. Click **Save**.
4. In similar way, the mentioned fields can be made mandatory for Bill To and End User.
1.3.3.3.5 Pricing/ Discounting Setup

Do the following steps to setup the pricing and discount:

Create Markets

Purpose

A market is created in SAP CPQ against every Sales Organization of SAP S/4HANA system. This will help in pricing determination in Sales Quote in CPQ system.

Procedure

1. Login to your SAP CPQ system using an Administrator User.
2. Click on SETUP in menu bar and choose Setup.
3. In left navigation bar, select Markets under Pricing/Calculations.
4. Click on Add New Button.
5. Enter the Market Code field as a Sales Organization of S/4HANA system.
6. Enter Marketing Name, Currency, and Market Factor.

<table>
<thead>
<tr>
<th>Market Code</th>
<th>Market Name</th>
<th>Currency</th>
<th>Market Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1710</td>
<td>Sales Org US</td>
<td>USD</td>
<td>1</td>
</tr>
<tr>
<td>1010</td>
<td>Sales Org DE</td>
<td>EUR</td>
<td>1</td>
</tr>
</tbody>
</table>

![Market Code Form]

SAP CPQ Integration Guide
Integrations with SAP Solutions
7. Click Save.
8. In left navigation bar, select Market Visibility under Pricing/Calculations.
9. Click on Add New Button.
10. Select Market Name created in above step in Step 2 Section.
11. In Step 3 section enter Visibility Condition as 1.
12. Click Save.

**Create Pricebooks**

**Purpose**

Pricing in the Sales Quote is determined based on market, pricebook and product in CPQ system. Pricebook comprises of Distribution Channel and Product entries with their respective prices.

**Procedure**

1. In left navigation bar, select Pricebooks under Pricing/Calculations.
2. Under Pricebooks tab, search for the Market Name created in above steps.
3. Click on Add New Pricebook button.
4. Enter the Pricebook name.
5. Enter the Distribution Channel similar to the distribution Channel of S/4HANA system against the Sales Organization. Example: 10
6. Check the Visible to everyone checkbox.
7. Click **Save**.
8. Pricing for the products in **Entries** tab will be created by Pricing Replication IDoc from S/4HANA system.

**Discounting**

**Purpose**

One of the major functionalities of SAP CPQ is on the fly discounts based on products, product types, product categories, users and user types. You can setup discounting in Sales Quote in your SAP CPQ system following below procedure.

**Procedure**

1. In left navigation bar, select **Discounts/Multipliers** under **Pricing/Calculations**.
2. Click on **Add New** button.
3. Click on Select button to maintain **USER TYPE as Sales** in Step 1 section.

   **Note**

   The above selection will allow users with user type ‘Sales’ to give discounts in Sales Quote which we will be defining in next step. You can create any condition for discount selection with User, User Type, Product, Product Type or Category.

4. In Step 2 section, maintain multiplier as 1 for **Minimal Value**, **Default Value**, and **Maximum Value**.
5. Maintain **Maximum Value** for Discount as 10. It will allow your Sales User to give maximum of 10 percent discount. Above 10 percent will be subject to approval.
6. In Step 3 section, maintain *Condition* as 1. You can create conditions using scripting language also.
7. Click *Save*.

**Discounting Priorities**

**Purpose**

In SAP CPQ, you can provide discount to customer in Sales Quote based on User, User Type, Product, Product Type or Product Category conditions. There might come a case with two or more valid conditions. In such situation, which condition will take priority.

**Procedure**

1. In left navigation bar, select *Discounting Priorities* under *Pricing/Calculations*.
2. As we want Product to have highest Priority. Select *Rank* as 1.
3. Further Select *Rank* for *Product Type* as 2, *Category* as 3, *User* as 4, and *User Type* as 5.

<table>
<thead>
<tr>
<th>Field</th>
<th>Field Label</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Type</td>
<td>User Type</td>
<td>5</td>
</tr>
<tr>
<td>User</td>
<td>User</td>
<td>4</td>
</tr>
<tr>
<td>Category</td>
<td>Category</td>
<td>3</td>
</tr>
<tr>
<td>Product</td>
<td>Product</td>
<td>1</td>
</tr>
<tr>
<td>Product Type</td>
<td>Product Type</td>
<td>2</td>
</tr>
</tbody>
</table>

4. Click *Save*.
1.3.3.3.6 Sales Quote to Order Setup

Do the following steps:

Add Action for External Sales Order in ERP

**Purpose**

The Action for creating Sales Order in your ERP External System and the same need to be assigned to Sales Quote in SAP CPQ.

**Procedure**

1. In left navigation bar, select *Workflow* under *Workflow/Approval*.
2. In *My Quotes* tab, Click on Start Status is Preparing, and End Status is Order Confirmation pending.

3. Click on , Scroll to the *Place order to ERP* Action. Click the checkbox *Selected*.
4. Click *Save*.
5. In a similar way configure all the below actions in workflow.

<table>
<thead>
<tr>
<th>Start Status</th>
<th>End Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Confirmation pending</td>
<td>Order Confirmation pending</td>
<td>Edit, Place Order to ERP, Save Quote</td>
</tr>
<tr>
<td>Order Confirmation pending</td>
<td>Order Placed</td>
<td>Order Status Update</td>
</tr>
<tr>
<td>Order Placed</td>
<td>Order Placed</td>
<td>Edit, History</td>
</tr>
</tbody>
</table>
Copy the same settings from *My Quotes* to *Other Quotes* tab as well. To do this Select *Other Quotes* tab. Select *My Quotes* from the dropdown selection in the Copy from field and click on *Copy* button.

6. In left navigation bar, select *Workflow Permissions* under *Workflow/Approval*.
7. Under Action Permissions, search for *Place order to ERP* action.
8. Click on Add button (+). And Select the checkbox for User Type Sales.

![Workflow Permissions » Edit of Place order to ERP](image)

9. Click Save.
10. In a similar way add the User Type Sales for all the actions like Edit, Save Quote, Order Status Update and History under Action Permissions.

Create Custom field in Sales Quote

**Purpose**

Custom field is needed in SAP CPQ in Sales Quote to capture PO Number in Sales Order.

**Procedure**

1. Login to your SAP CPQ system using an Administrator User.
2. Click on SETUP in menu bar and choose Setup.
3. In left navigation bar, select Custom Fields under Quotes.
4. Click on Add New Button.
5. In the Definition tab, maintain the Name as PO Number (case sensitive).
6. Choose Type as Free Form and Tab as Additional Info.
7. Select Save quote on change.
8. Enter **Default Label** as **PO Number**.

```
Custom Fields » Edit of PO Number

<table>
<thead>
<tr>
<th>Definition</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Name</td>
<td>PO Number</td>
</tr>
<tr>
<td>*Type</td>
<td>Free Form</td>
</tr>
<tr>
<td>Tab</td>
<td>Additional Info</td>
</tr>
<tr>
<td>Order within Tab</td>
<td>6</td>
</tr>
<tr>
<td>Save quote on change</td>
<td>√</td>
</tr>
<tr>
<td>Calculation Type</td>
<td>None</td>
</tr>
<tr>
<td>Transfer to items</td>
<td>☐</td>
</tr>
<tr>
<td>Show on place order confirmation page</td>
<td>☐</td>
</tr>
<tr>
<td>Default Label</td>
<td>PO Number</td>
</tr>
</tbody>
</table>
```

9. Click **Save**.

### 1.3.3.4 Initial Data Load

This section describes how to execute the initial data load of material, customer master and Pricing Condition data from your SAP S/4HANA system to the SAP CPQ system.

### 1.3.3.4.1 Business Partner (Customer Master)

1. Open transaction **DRFOUT**.
2. Select the **Replication Model** and **Outbound Implementation** created in S/4HANA section.
3. Choose **Replication Mode** as **Manual**, in case you want to replicate specific business partners.

5. Enter the Business Partner ID and choose Back.

6. Choose Yes in the following pop-up.

7. Choose Execute.
1.3.3.4.2 Product Master

1. Open transaction DRFOUT.
2. Select the Replication Model and Outbound Implementation created in S/4HANA section.
3. Choose Replication Mode as Manual, in case you want to replicate specific Material.

5. Enter the Material (For example: TG11) and choose Back.

6. Choose Yes in the following pop-up.
7. Choose **Execute**.

### 1.3.3.4.3 Product Pricing

1. Access the transaction SE38.
2. Enter the report name RCOND_CONDITIONS_EXTRACT and click on Execute.

3. Enter the following details and click **Execute**:

4. Enter the following details:
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Organization</td>
<td>For example: 1000</td>
</tr>
<tr>
<td>Distribution Channel</td>
<td>For example: 10</td>
</tr>
<tr>
<td>Valid from</td>
<td>A valid date</td>
</tr>
<tr>
<td>Valid from</td>
<td>A valid date</td>
</tr>
<tr>
<td>Logical System</td>
<td>&lt;Logical system of the SAP CPQ&gt;</td>
</tr>
<tr>
<td>Test Mode</td>
<td>Unchecked</td>
</tr>
<tr>
<td>Table</td>
<td>304(Material with release status)</td>
</tr>
</tbody>
</table>

5. The condition records created with the provided data are replicated to SAP CPQ.

6. Execute the report by changing the scale type to B (retaining the other inputs) for which IDoc filtering is not maintained.

    **Note**
    
    Go to Tcode BD87 in SAP S/4HANA system to process the Idoc if it is not triggered automatically.

### 1.3.4 Manual Build Steps

#### 1.3.4.1 Ticket Component

<table>
<thead>
<tr>
<th>Implementation Step</th>
<th>Component</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SV-RDS-CFC</td>
<td></td>
</tr>
</tbody>
</table>

#### 1.3.4.2 Appendix
1.3.4.2.1 How to find CPQ API URL

**Purpose**

You need to have CPQ API URL, which will be used in Cloud Platform Integration system as address path to call the API's.

**Procedure**

1. Login to your SAP CPQ system using an Administrator User.
2. Click on SETUP in menu bar and choose Setup.
3. On the top right corner, click on Help icon.
4. On left navigation, click on CallidusCloud CPQ API.
5. Check the API Example section.
6. To access CPQ's API, use the following URLs:
   - `<application root>/wsAPI/CPQAPI.asmx` - this is the set of user level functions like Business Partner Replication and Sales Order Confirmation.
   - `<application root>/wsAPI/wssrv.asmx` - this is the set of admin level functions like Material Replication and Pricing Condition Replication.

1.3.4.2.2 How to setup Cloud Connector

**Purpose**

You need to have Cloud Connector Virtual Host URL for SAP S/4HANA system, which will be used in Cloud Platform Integration system as address path to call the S/4HANA system.

**Procedure**

1. Login to your SAP HANA Cloud Connector system using an Administrator User.
2. Select the Account of your Cloud Platform Integration system.
3. Click on Access Control to add Mapping Virtual to Internal System.
4. Click on Add button.
5. Select ABAP System in Back-end Type and Click Next.
6. Select HTTPS as Protocol and Click Next.
7. Enter the Internal Host and Click Next. You can find HTTPS host and port of your system in SMICM transaction Goto Services.
8. Enter the Internal Port and Click Next.
9. Enter the Virtual Host which you must maintain in Cloud Platform Iflows and Click Next.
10. Enter the Virtual Port which you must maintain in Cloud Platform Iflows and Click Next.
11. Select Principal Type as none and Click Next.
12. Enter the Description and Click Next.
13. Click on Finish button.

14. Select the Entry for Virtual Host Mapping created in above steps.
15. Click on Add in Resource Accessible and Add / as URL Path.

16. Click Save.
1.4 Variant Configuration Integration

The SAP CPQ – Variant Configuration integration unites front and back-office processes to simplify complex product configurations for sales representatives.

The SAP CPQ – Variant Configuration integration is available for both the Quote 1.0 and Quote 2.0 engines. The integration works in the same way on both quote engines, the only difference being that in the integration for the Quote 2.0 engine, the Copy Quote and Place Order to ERP actions are currently not supported. Fixing these two limitations is on the team’s roadmap and will be handled with priority in the upcoming releases.

The Variant Configuration integration is provided by the configuration and pricing services of SAP Variant Configuration and Pricing. Limitations of the Variant Configuration integration with SAP CPQ are available in SAP Note 2726240.

Introduction to the Variant Configuration Integration [page 78]
Integrating SAP CPQ with Variant Configuration provides a cloud-based solution for simplifying and enhancing sales of complex configurable products.

Integrate SAP CPQ with SAP Variant Configuration and Pricing [page 82]
To enable the SAP Variant Configuration and Pricing integration on the SAP CPQ side, configure the following settings in the Setup.

Define SAP General Attribute Mappings [page 86]
After connecting SAP CPQ to SAP Variant Configuration and Pricing, you need to map general attributes.

Technical Overview [page 91]
The Variant Configuration integration is provided by the SAP Variant Configuration and Pricing service.

Pricing [page 98]
You can select Variant Pricing or Pricebook Lookup as the pricing type for Variant Configuration products in SAP CPQ.

1.4.1 Introduction to the Variant Configuration Integration

Integrating SAP CPQ with Variant Configuration provides a cloud-based solution for simplifying and enhancing sales of complex configurable products.

Why Integrate SAP CPQ with Variant Configuration

Variant Configuration is used for configuring exceedingly complex products. Although SAP CPQ also supports product configuration, Variant Configuration exists specifically for configuring complex tangible goods for manufacturing purposes.

In other words, although Variant Configuration models can be recreated in SAP CPQ, that would come at a high cost in terms of time and resources. That is why the primary motivation for integrating SAP CPQ with Variant Configuration is eliminating these costs. This is especially convenient to clients who are already working with Variant Configuration models and wish to begin using SAP CPQ.
Moreover, highly complex models are typically a back office feature. However, an integration unites the front office practicality of SAP CPQ with the detailed features of Variant Configuration, resulting in a well-rounded solution.

On the one hand, integrating Variant Configuration expands the technical capabilities of SAP CPQ. On the other hand, SAP CPQ makes Variant Configuration models client-friendly and sales-oriented. With an integrated solution, users can experience Variant Configuration models in a practical, fast and lightweight cloud context, featuring a user-friendly and customizable interface. In SAP CPQ, the models can be adapted for a specific market or customer group, thus further optimizing the sales process.

In other terms, integrating SAP CPQ with Variant Configuration provides a cloud-based solution for simplifying and enhancing sales. However, this integration is unrelated to the manufacturing process, which primarily relies on on-premise solutions.

**Note**

Variant Configuration integration is available for both the Quote 1.0 and Quote 2.0 engines. The integration works in the same way on both quote engines, the only difference being that in the integration for the Quote 2.0 engine, the Copy Quote and Place Order to ERP actions are currently not supported. Fixing these two limitations is on the team's roadmap and will be handled with priority in the upcoming releases.

**Integration Flow**

First, SAP ERP supplies SAP CPQ with simple materials via SOAP API. In SAP CPQ, products are imported from SAP ERP with specific product data (see Technical Overview [page 91] for the list of data that is sent) and flagged as products originating from SAP systems. This is performed via SAP Cloud Platform Integration, a cloud-based microservice which facilitates the integration of front office and back office applications.

Once simple products are imported from SAP ERP, SAP CPQ needs to obtain the configuration in order for these products to become complex, configurable Variant Configuration models. This information is obtained through the process of knowledge base synchronization – the process of exporting configurations and other model-related information from back office applications into the cloud.
A knowledge base contains relevant information such as configurations, characteristics and pricing conditions. When knowledge base synchronization is executed, products flagged as SAP products are converted into configurable or parent/child products. The models are available in the product catalog and can be configured and added to the quote. When these products are configured on the user side, rules and constraints are triggered from the SAP Variant Configuration and Pricing configuration endpoint.
SAP Variant Configuration and Pricing [page 82] is the service providing this integration. It provides SAP CPQ with knowledge bases, the configuration result and the pricing result.

When SAP CPQ is integrated with Variant Configuration, the Quote-First Approach [page 95] is used. The reason for that is the fact that the market and other fields need to be selected for SAP Variant Configuration and Pricing to use reference characteristics and pricing attributes [page 90] in order to provide SAP CPQ with the correct configuration.

An example of a reference characteristic is customer country – if this value is specified in SAP CPQ, the characteristic provides SAP Variant Configuration and Pricing with the context that is necessary for it to retrieve the correct response from the pricing table.

For technical details about the integration flow, refer to the Technical Overview [page 91].

Related Information

System Requirements for SAP Integrations [page 5]
1.4.2 Integrate SAP CPQ with SAP Variant Configuration and Pricing

To enable the SAP Variant Configuration and Pricing integration on the SAP CPQ side, configure the following settings in the Setup.

Prerequisites

- The application parameter *Allowed origins for the CORS filter* must be configured in the way explained in Application Parameters.

Context

All the information needed to connect SAP CPQ with the SAP Variant Configuration and Pricing service should be provided by the administrator managing SAP Variant Configuration and Pricing. Settings are configured on the tenant level.

SAP Variant Configuration and Pricing onboarding and provisioning must first be completed in the manner outlined in the SAP Variant Configuration and Pricing administration guide.

i Note

To exchange variant configuration and pricing data between cloud solution SAP CPQ and the licensee’s local on-premise SAP ERP back end, the SAP Variant Configuration and Pricing provider also comprises SDI Data Provisioning Agent (part of SAP HANA smart data integration). SDI Data Provisioning Agent is an on-premise application that is to be installed and operated locally in the customer’s on-premise landscape.

For data exchange with SAP back-end systems using a third-party runtime database obtained from SAP (such as Microsoft SQL Server, IBM DB2, Oracle), certain restrictions apply according to the applicable on-premise license agreements for the third-party runtime databases. One such restriction is that the licensee may not access these databases directly via systems or technologies that have not been licensed specifically for use with such runtime database. Therefore, the corresponding SDI Data Provisioning Agent LogReader Adapters provided with the SAP Variant Configuration and Pricing provider for SAP CPQ may not be used to directly access, or replicate data from, such third-party runtime databases licensed from SAP. For third-party database licenses not obtained from SAP, please refer to the respective license agreements on the available options for data exchange via direct database accesses.

For more information about interoperability and maintenance rules for Data Provisioning Agent (DP Agent), see SAP Note 2546811.
Procedure

1. Go to Setup > Providers > Providers > SAP.
   Available SAP providers display.
2. Click SAP Variant Configuration and Pricing.
   A new page displays.
3. Set Enable SAP Variant Configuration and Pricing to TRUE.
   When enabled, your current tenant is able to communicate with remote configuration and pricing services.
4. In Authentication Settings, provide the following data:
   a. Authentication URL: the URL of the SAP Variant Configuration and Pricing authorization service. This URL must contain the full path, including /oauth/token.
   b. Configuration Client Identifier: OAuth 2.0 client ID issued by the SAP Variant Configuration and Pricing authorization provider, used when obtaining products from SAP Variant Configuration and Pricing.
   c. Configuration Client Secret: OAuth 2.0 client secret issued by the SAP Variant Configuration and Pricing authorization provider, used when obtaining products from SAP Variant Configuration and Pricing.
   d. Pricing Client Identifier: OAuth 2.0 client ID issued by the SAP Variant Configuration and Pricing authorization provider, used when obtaining pricing from SAP Variant Configuration and Pricing.
   e. Pricing Client Secret: OAuth 2.0 client secret issued by the SAP Variant Configuration and Pricing authorization provider, used when obtaining pricing from SAP Variant Configuration and Pricing.
5. In Base URLs, provide the root SAP Variant Configuration and Pricing API URLs. The root URL pattern is: https://appname.cfapps.region.hana.ondemand.com, where appname is either cpservices-product-configuration (for the configuration service) or cpservices-pricing (for the pricing service) and region is eu10, us10, ap11.
   a. Configuration Base URL: the root URL of configuration services. See the sample endpoints for the configuration root URL in the SAP API Business Hub.
   b. Pricing Base URL: the root URL of pricing services. See the sample endpoints for the pricing root URL in the SAP API Business Hub.
6. In Endpoints, define the following SAP Variant Configuration and Pricing API endpoints:
   a. Configurations Endpoint: relative endpoint to the API that handles configuration creation and update operations. Enter: /api/v2/configurations.
   b. External Configurations Endpoint: relative endpoint that manages external configurations and configuration deep copies. Enter: /api/v2/externalConfigurations.
   c. Knowledge Base Determination Endpoint: relative endpoint to the API that handles and resolves a specific knowledge base (a collection of all relevant product model data, such as materials, characteristics and dependencies) by retrieving the complete knowledge base content. Enter: /api/v2/kbdetermination.
   d. Knowledge Base Endpoint: relative endpoint to the API that retrieves the list of knowledge bases that were changed. Enter: /api/v2/knowledgebases.
   e. Pricing Procedure Endpoint: relative endpoint used for pricing attributes and determination details based on the provided pricing procedure name. Enter: /api/v1/pricingprocedure.
   f. Stateless Pricing Endpoint: relative endpoint to the API that retrieves price and pricing details for the provided configuration and pricing procedure. Enter: /api/v1/statelesspricing.
7. See Knowledge Base Synchronization Settings [page 84] for additional knowledge base synchronization specifications.

8. Save your changes. A confirmation message displays.

9. Click *Execute Job*.

When integrating SAP Variant Configuration and Pricing for the first time, you need to save all the changes you made in order to be able to click the *Execute Job* button. Clicking the *Execute Job* button runs the knowledge base synchronization immediately. Once the synchronization starts, its status (**Started***) displays below the button. The next time you execute a job, the date of the last synchronization will also be displayed.

The first time knowledge base synchronization is executed, all knowledge base details from SAP Variant Configuration and Pricing which correspond to products and attributes in SAP CPQ are replicated in SAP CPQ. Every next time knowledge base synchronization is executed, only the information that was changed after the date of the last knowledge base synchronization will be updated. If the first synchronization job fails to execute properly, please contact the SAP CPQ Support team to identify and resolve the issue.

**Next Steps**

After configuring the settings on this page, you need to **Define SAP General Attribute Mappings [page 86]**.

Knowledge Base Synchronization Settings [page 84]

In Knowledge Base Synchronization Settings, you can set up the knowledge base synchronization language and translations, product layout source, notification recipient email address, daily execution start time, as well as execute jobs from a past date.

**1.4.2.1 Knowledge Base Synchronization Settings**

In **Knowledge Base Synchronization Settings**, you can set up the knowledge base synchronization language and translations, product layout source, notification recipient email address, daily execution start time, as well as execute jobs from a past date.

- **Knowledge Base Synchronization Language** - the default languages used during knowledge base synchronization. Knowledge base details such as product name, product description and attribute data are retrieved in the specified languages. Administrators who maintain translations for their models in SAP Variant Configuration and Pricing can synchronize these translations with the products in SAP CPQ thanks to this feature. Every time knowledge base synchronization is executed, translations for the specified languages are synchronized for all products and product versions obtained via that specific knowledge base synchronization. Languages can be specified by using two-letter ISO language codes (for example **DE**, **EN**) separated by a comma. However, before languages can be specified on this page, language ISO codes need to be defined for individual **dictionaries**. If the knowledge base synchronization language is not provided, translations are not synchronized.

- **Add Translations for Synchronized Products** - when the checkbox is selected, during the next knowledge base synchronization, translations for all languages specified in the field above are added both for products obtained via the ongoing synchronization, and for products that were obtained via previous
synchronizations. Once the translations are synchronized retroactively in this manner, the checkbox is automatically unselected.

Translations obtained via knowledge base synchronization behave in the following manner:

- If a translation is not successfully synchronized for one product, it does not affect the synchronization of translations for other products.
- If knowledge base synchronization is successful, but a translation is not successfully synchronized for one product, only the product that did not synchronize successfully is synchronized again during the following job.
- If knowledge base synchronization is not successful, no translations are synchronized.
- Translations for tab names are synchronized during the initial knowledge base synchronization. In every subsequent job, new translations are only added for the tab names that are already synchronized. Translations are not synchronized for the names of any new tabs that were subsequently added.

**Always Use Product Layout from Back-End Configurator** - if this parameter is set to **TRUE**, product layouts are automatically updated to match the layout from the back-end configurator with every knowledge base synchronization. As a result, it is only necessary to manually maintain the layout of the back-end model. If set to **FALSE**, the back-end product layout is synchronized only during the first knowledge base synchronization. In that case, you can continue to manually update the product layout in SAP CPQ. The default value is **FALSE**. To update the layout of previously synchronized products so that it matches the layout from the back-end configurator, these products would need to be synchronized again with the parameter value set to **TRUE**.

**Notification Recipient Email** - the email address to which notifications about knowledge base synchronization failure are sent. The notification contains problematic knowledge bases and diagnostic details. Additionally, the notification can be sent to multiple recipients, in which case you need to separate the email addresses by commas.

**Daily Execution Start Time (UTC)** - a 24-hour time format used to schedule knowledge base synchronization on a daily basis. We recommend that you select your customers’ non-business hours.

**Execute Job Immediately** - you can execute a knowledge base synchronization job at any given time. Additionally, by selecting a date in the **Sync from date** field, you can repeat a knowledge base synchronization which was performed in the past. When the next job is executed, SAP CPQ is updated with any changes that were made to knowledge bases on or after the specified date. You can’t select a date later than the last executed job. Consequently, you can now repeat jobs which were initially incomplete or affected by errors. For example, if a job is mistakenly recorded as successful, data is missing because systems ran in different time zones, or previously synchronized data was changed and needs to be updated, you can repeat the job in SAP CPQ to quickly retrieve the information.

**Related Information**

Integrate SAP CPQ with SAP Variant Configuration and Pricing [page 82]
Technical Overview [page 91]
1.4.3 Define SAP General Attribute Mappings

After connecting SAP CPQ to SAP Variant Configuration and Pricing, you need to map general attributes.

Procedure

1. Go to Setup > Providers > Providers > SAP.
   
   Available SAP providers display.

2. Click SAP General Attribute Mappings.
   
   A new page displays.

3. Under Global Attributes, fill out the following fields:
   
   a. (Optional) In Country Code, enter a two-character country code (for example, US).

      A country code should be used only if Variant Configuration runs product rules that rely on the country code. Additionally, if you need to dynamically build a country code, you can do so by clicking to build a formula. When using a CTX tag, the tag also needs to return two characters.

      This is not a mandatory setting. This value is used for the execution of product rules and pricing procedure evaluation in SAP Variant Configuration and Pricing. If it is not used for either of those, please skip this setting.

   b. In Division, enter a two-character code representing the SAP ERP division (for example, 01). Click to build a formula to dynamically retrieve the value for this field.

4. Under Pricing Procedure, fill out the following fields:
   
   a. In Pricing Procedure Name, enter the name of the pricing procedure (for example, ZRt52t), which you obtained from SAP ERP. See Set Up Pricing Procedure in SAP ERP [page 100].

      This pricing procedure will be used for calculating pricing for configurable products that are synchronized between SAP ERP and SAP Variant Configuration and Pricing. Additionally, if you need to dynamically build a pricing procedure name, click to build a formula.

   b. In Pricing Procedure Name for Catalog Prices, enter the name of the pricing procedure used for obtaining base price values for synchronized products and displaying them in the Catalog. In a typical use case, this pricing procedure contains the base price calculation. It is less complex than the pricing procedure used for obtaining both the base price and variant condition values. Setting up a pricing procedure and maintaining it in SAP ERP is explained in Set Up Pricing Procedure in SAP ERP [page 100].

   c. In Base Price, enter the base price value or click to build a formula that dynamically obtains the value. The default Base Price value is BASE. This value also needs to be maintained in SAP ERP. See Set Up Pricing Procedure in SAP ERP [page 100], where the example value ZSS1 is used instead of BASE.

   d. In Selected Options, enter the condition function used for calculating the breakdown and the total of the selected price-related options. Additionally, if you need to dynamically build the value for this field,
click to build a formula. The default value is VARC. This value also needs to be assigned to variant condition types in SAP ERP. See Set Up Pricing Procedure in SAP ERP [page 100].

e. In Automatically Reconfigure on Pricing Attributes or Pricing Procedure Change, if TRUE is selected, whenever the following entities are changed in a quote, SAP CPQ checks if the pricing procedure or pricing attributes for the quote items have changed:

○ customers
○ customer fields
○ quote custom fields
○ quote item custom fields

If they are different, the quote items are reconfigured. For more information, see Calculation Pipeline for Variant Configuration Products [page 87].

5. Under Reference Attributes and Pricing Attributes, you can edit the standard reference characteristics used in integrations with SAP systems, or create custom reference and pricing attributes. See Reference Characteristics and Pricing Attributes [page 90].

\[\text{Note}\]

To be able to configure products without having to create a quote first, see Configuring Products Without First Creating a Quote [page 89].

6. Click Save.

The mapped attribute values are now in use.

Calculation Pipeline for Variant Configuration Products [page 87]
Reference and pricing attributes on products synchronized from SAP Variant Configuration and Pricing are evaluated in a specific order.

Configuring Products Without First Creating a Quote [page 89]
By default, with the Variant Configuration integration enabled, a quote needs to be selected before you can configure a product, because the configuration obtains reference and pricing attributes from the currently selected quote. However, it is possible to set up a separate pricing procedure and set of attributes, used only when a product is being configured without first creating a quote.

Reference Characteristics and Pricing Attributes [page 90]
Standard reference and pricing attributes used by SAP CPQ for the SAP Variant Configuration and Pricing integration are listed on this page. You can edit reference and pricing attributes or create custom reference characteristics in SAP General Attribute Mappings.

1.4.3.1 Calculation Pipeline for Variant Configuration Products

Reference and pricing attributes on products synchronized from SAP Variant Configuration and Pricing are evaluated in a specific order.

The following is an example of a formula used for Variant Configuration products. This is a formula for pricing attributes that should retrieve a value from a custom table based on a product material number (SystemId),
sales organization and distribution chain, and, if the returned value is empty, use the value from the base product table instead.

```sql
<* ISNULL(<* TABLE( SELECT UoM FROM Materials WHERE SalesOrg = '<*CTX( Market.Code )*>' and DistributionChannel = '<*CTX( Pricebook.DistributionChannel )*>' and MaterialId = '<*CTX( Product.SystemId )*>' ) *>, <*CTX( Product.UnitOfMeasure )*>) *>
```

**Evaluation of Attributes Upon Creating a Request**

All reference attributes are evaluated at the root product level (root product logic is used). Pricing attributes are evaluated at the root item product level, or at both the root and the child item product level if the *Evaluated per item* checkbox is selected for the attribute in question in *SAP General Attribute Mappings* (see *SAP General Attribute Mappings* [page 90]). Tags such as `<*CTX( Market.Code )*>` and `<*CTX( Pricebook.DistributionChannel )*>` can be evaluated in the product context and have the same value for all items, while the tag `<*CTX( Product.SystemId )*>` is evaluated as a different value for each item.

**Obtaining New Prices Upon Pricing Attribute or Pricing Procedure Change**

When a user edits a quote, SAP CPQ checks whether pricing attributes or the pricing procedure are changed. If they are changed, the *Reconfigure* action is executed in order to retrieve new prices from SAP Variant Configuration and Pricing. If the pricing procedure or pricing attributes evaluated at the quote header level are changed, the *Reconfigure* action is executed for all SAP Variant Configuration and Pricing root items. This is because they affect prices for all items. If a change is detected on the value of a pricing attribute which is evaluated on the item level, the affected root item is reconfigured.

The check and the subsequent Reconfigure action are performed only if *Automatically Reconfigure on Pricing Attributes or Pricing Procedure Change* is enabled in *General Attribute Mappings*. The check is performed when the *Reprice (Calculate)* action is performed. *Reprice (Calculate)* is performed when the user changes a quote item field, a quote custom field triggers saving, or when the *Save Quote* action is executed. The check is also performed when customer selection is changed or when a customer field is changed.

The pricing request which is sent to SAP Variant Configuration and Pricing is captured (saved in memory and to database) for each SAP Variant Configuration and Pricing root item. When the check is performed, the pricing attributes and the pricing procedure name are evaluated again and compared to the corresponding data from the previous pricing request. If a change affecting all items is detected (pricing procedure name change or header-level pricing attribute change), the check is stopped and *Reconfigure* is executed for all SAP Variant Configuration and Pricing root items. Otherwise, item-level pricing attributes are evaluated and compared for each root item and its descendants. If a change is detected, the affected root items are reconfigured.

During the check, the attribute formulas are evaluated in the quote context, while during pricing request creation, they are evaluated in the product context. It is possible that some formulas that can be evaluated in the product context, such as product based CTX tag formulas, cannot be evaluated in Quote context, and therefore have empty values. If there are such attributes (whose value is evaluated as empty in the quote context, but they were not empty in the previous pricing request) they are not regarded as changed during the check, and they do not cause *Reconfigure*, in order for the system to avoid false positives and avoid executing...
the Reconfigure action too frequently. Additionally, you can always execute the Reconfigure action manually on the quote.

Related Information

Reference Characteristics and Pricing Attributes [page 90]
Define SAP General Attribute Mappings [page 86]
Technical Overview [page 91]

1.4.3.2 Configuring Products Without First Creating a Quote

By default, with the Variant Configuration integration enabled, a quote needs to be selected before you can configure a product, because the configuration obtains reference and pricing attributes from the currently selected quote. However, it is possible to set up a separate pricing procedure and set of attributes, used only when a product is being configured without first creating a quote.

Users may use this feature to explore the configuration options of a new model before selling it to a customer, without having to create an empty quote first to initiate the configuration process. After activating the toggle switch Configure products without first creating a quote, the SAP General Attribute Mappings page is split into two tabs. The original set of attributes is located in the Attributes (with Created Quote) tab. The second tab Attributes (No Quote) lists the settings that are used if there is no quote context available.

In Attributes (with Created Quote), the administrator can refer to the quote context when specifying the reference and pricing attributes. This is not possible for Attributes (No Quote), as there is no existing quote context. Regardless, the administrator should provide some meaningful defaults in this case, which can be fixed values or values retrieved from a custom table through formulas defined in the Formula Builder. The defaults chosen here depend on the specifics of your business: for example, if most products are sold to customers from a specific country, it should be set as the default country. If you are mainly trading via the Internet, you likely want to set the corresponding division as default. Typically, a minimal set of reference characteristics required to run the configuration should be specified here – in most cases, it should be enough to specify the country, division, and pricing procedure. The defaults used for configuration can be synchronized with any defaults used in the quote process.

Related Information

Define SAP General Attribute Mappings [page 86]
1.4.3.3 Reference Characteristics and Pricing Attributes

Standard reference and pricing attributes used by SAP CPQ for the SAP Variant Configuration and Pricing integration are listed on this page. You can edit reference and pricing attributes or create custom reference characteristics in SAP General Attribute Mappings.

Any reference characteristic from SAP Variant Configuration and Pricing can be mapped to any custom field in SAP CPQ. Attributes defined in the Reference Attributes section are used for sending reference characteristics to SAP Variant Configuration and Pricing with every configuration request. Similarly, attributes defined in the Pricing Attributes section are sent with every pricing request. Attribute values (defined on the SAP CPQ side) can be either hard-coded or in the form of formulas.

Evaluating Pricing and Reference Attributes Per Item

Pricing and reference attributes are evaluated in the context of the quote header information by default. However, they can also be evaluated in the context of quote items. You can enable that by editing a pricing attribute and selecting the checkbox Evaluated per item in General Attribute Mappings, pricing attributes for which this checkbox is selected are marked as True in the Evaluated per item column. As a result, tags such as <<CTX(Product.SystemId)>>, <<CTX(Product.UnitOfMeasure)>>, <<CTX(Product.PartNumber)>> can be used in attribute formulas, and therefore, a pricing attribute or a reference attribute value can differ from item to item. For more information, see Calculation Pipeline for Variant Configuration Products [page 87].

Reference Characteristics

SAP CPQ uses the following reference characteristics (attributes) for the configuration context:

<table>
<thead>
<tr>
<th>Context Keys</th>
<th>Context Values</th>
<th>Match in SAP CPQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBPA_AG-LAND1</td>
<td>SAP country code (two-character code)</td>
<td>Country code</td>
</tr>
<tr>
<td>VBPA_RG-LAND1</td>
<td>SAP country code (two-character code)</td>
<td>Country code</td>
</tr>
<tr>
<td>VBAP-MATNR</td>
<td>SAP root material ID (internal format)</td>
<td>System Id</td>
</tr>
<tr>
<td>VBAK-KWMENG</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>VBAK-VTLEG</td>
<td>Distribution channel</td>
<td>Distribution channel in pricebooks</td>
</tr>
<tr>
<td>VBAK-VKORG</td>
<td>Sales organisation</td>
<td>Market code</td>
</tr>
<tr>
<td>VBAK-SPART</td>
<td>Division</td>
<td>Division</td>
</tr>
<tr>
<td>VBAK-KUNNR</td>
<td>SAP customer ID</td>
<td>Ship-to customer (for VBAK-KUNNR):</td>
</tr>
<tr>
<td>VBPA_AG-KUNNR</td>
<td>Bill-to customer (for VBPA_AG-KUNNR and VBPA_RG-KUNNR)</td>
<td></td>
</tr>
<tr>
<td>VBPA_RG-KUNNR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VBAK-ERDAT</td>
<td>Creation date</td>
<td>Effective date</td>
</tr>
</tbody>
</table>
Pricing Attributes

The following attributes are used when sending the price inquiry over SAP Variant Configuration and Pricing:

<table>
<thead>
<tr>
<th>Level</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item (product)</td>
<td>KOMP-PMATN</td>
<td>Material number</td>
</tr>
<tr>
<td></td>
<td>KOMP-SPART</td>
<td>Division</td>
</tr>
<tr>
<td></td>
<td>KOMP-PRSFD</td>
<td>Set to X</td>
</tr>
<tr>
<td>Header (business partner) - Quote, Customer in CPQ</td>
<td>KOMK-VKORG</td>
<td>Sales organization</td>
</tr>
<tr>
<td></td>
<td>KOMK-VTWEG</td>
<td>Distribution channel</td>
</tr>
<tr>
<td></td>
<td>KOMK-SPART</td>
<td>Division</td>
</tr>
<tr>
<td></td>
<td>KOMK-LAND1</td>
<td>Country key</td>
</tr>
<tr>
<td></td>
<td>KOMK-KUNNR</td>
<td>Customer code</td>
</tr>
<tr>
<td></td>
<td>KOMK-WAERK</td>
<td>Currency code of the selected market</td>
</tr>
<tr>
<td></td>
<td>KOMK-PRSDT</td>
<td>Quote date created</td>
</tr>
<tr>
<td></td>
<td>KOMK-FBUDA</td>
<td>Quote date created</td>
</tr>
</tbody>
</table>

Related Information

Calculation Pipeline for Variant Configuration Products [page 87]
Define SAP General Attribute Mappings [page 86]
Technical Overview [page 91]

1.4.4 Technical Overview

The Variant Configuration integration is provided by the SAP Variant Configuration and Pricing service. First, master material is sent from SAP ERP to SAP Cloud Platform Integration over SAP IDocs. SAP Cloud Platform Integration then calls the SAP CPQ SOAP API, and the following product data is sent from SAP ERP to SAP CPQ:

- name
- description
- part number
- material number (which becomes system ID in SAP CPQ)
- unit of measure
- catalog/hierarchy (which becomes category in SAP CPQ)
- material type (which becomes product type in SAP CPQ)

Products that are imported from SAP ERP in this way become simple products in SAP CPQ and are placed in the preselected category. If there is a product hierarchy in SAP ERP, the hierarchy will also be replicated in SAP.
CPQ as a category, which means that one product will belong to two categories – the category it belongs to in SAP ERP and the preselected category in SAP CPQ. If categories that exist in SAP ERP (also known as hierarchy) do not exist in SAP CPQ, they will be replicated in SAP CPQ. On the other hand, the preselected category needs to be manually created [page 9].

It is also possible to quantify products by entering an ISO code containing up to three characters in the Unit of Measure field in the Definition tab of a product in SAP CPQ. The maintained ISO code must be known in SAP ERP (see transaction CUNI). The Unit of Measure corresponds to the material's base unit of measure in SAP ERP. It is different from the sales unit of measure, since SAP CPQ does not support sales units that are different from the base unit. In Setup > Product Catalog > Products/Products 2.0, products obtained via synchronization with SAP ERP and SAP Variant Configuration and Pricing have the value Yes in the column Synced from Back Office. The column Synced from Back Office is visible only when SAP CPQ is integrated with a back office application, such as SAP ERP.

Furthermore, simple products previously imported from SAP ERP get their attributes and new product display type via knowledge base synchronization. During the first knowledge base synchronization, attributes are assigned to a specific characteristic group, which is a product tab in SAP CPQ. Attributes retain the same layout order they originally had in the Variant Configuration, with the exception of attributes appearing in more than one tab. Those attributes do not retain their original positions. With every new synchronization, new attributes will be added, but they will not be assigned to any product tab in SAP CPQ. This means that administrators will have to reorder attributes and assign them to appropriate product tabs by using the drag-and-drop functionality of the configuration layout in SAP CPQ. Attributes can’t be marked as line items for Variant Configuration products.

Consider this example: in SAP CPQ, there is a Hardware tab. The said tab is added to the following products: Monitor, Keyboard and Mouse. After some time, you decide to change the name of the Hardware tab to Gaming. Once the tab is renamed, the changes are visible in the aforementioned products, that is, the Gaming tab is now a part of Monitor, Keyboard and Mouse.

Additionally, SAP CPQ successfully handles synchronizing attributes (characteristics) with an empty or null possible value ID. If a characteristic with an empty or null possible value ID arrives to SAP CPQ via knowledge base synchronization, that value is not synchronized, but everything else is synchronized as expected. However, if that value is the only value in the characteristic, a default value is added, instead.

### i Note

SAP CPQ product tabs are globally defined, which means that any change to a tab name in SAP CPQ will affect all the products in which the tab is included. In SAP CPQ, the same tab, with the same ID, can be used in different products. However, in SAP Variant Configuration and Pricing, product tabs have IDs that start from 1 for each product.

On knowledge base synchronization, only the first tab with the ID 1 is created in SAP CPQ. The remaining tabs with the same ID are not created, and the first tab with that ID is used in their place. Only tabs with IDs that do not exist yet in SAP CPQ are created. To make sure tabs are used correctly in layouts after synchronization, you can assign unique names (IDs) to tabs in SAP Variant Configuration and Pricing. so that new tabs are created for new synchronized products. If you experience issues with incorrect layouts, contact the SAP CPQ Support team for assistance with synchronizing tabs.

Editing the quantity of a synchronized item (simple or configurable) in the quote triggers repricing and reconfiguration. New prices are retrieved from SAP Variant Configuration and Pricing. Therefore, SAP CPQ supports scale pricing, which is used in SAP Variant Configuration and Pricing.

When entering a numeric value for a free-input attribute synchronized from Variant Configuration, users can see the allowed range of values next to the input field. Any changes made to these value ranges are
synchronized and displayed in SAP CPQ in real time. SAP CPQ supports all types of value ranges that can be obtained from Variant Configuration.

Knowledge base synchronization is initiated from SAP CPQ. Depending on the knowledge base structure, the synchronization converts simple products to either configurable products or parent-child products by calling SAP Variant Configuration and Pricing to get the attribute name, description and value. As stated in Integrate SAP CPQ with SAP Variant Configuration and Pricing [page 82], knowledge base synchronization is run on a daily basis. In case the synchronization fails, the administrator is notified via email, and the job will be rescheduled for execution during the next synchronization. The job will get information about products that were created or updated in the last 24 hours. Additionally, every change to the product model in SAP ERP that results in a new knowledge base version will lead to a new product version. Product versions cannot be added or deleted in SAP CPQ. Additionally, product versions differ in effective date. Product rules are executed only in SAP Variant Configuration and Pricing.
CTX tags used in attributes on products in the quote are reevaluated every time one of the following actions is performed:

- Load Quote
- Change Market
- Change Customer
- Change Pricebook
- Save Quote
- Reconfigure

When CTX tags are reevaluated, the result is cached in the quote and pricing attributes are sent for each item. The product part number is added as another default attribute since there is a part number per item.

**i Note**

Configurable products synchronized from SAP Variant Configuration and Pricing to SAP Cloud for Customer can be used in a typical scenario [page 155] for that integration, where the user lands from SAP Cloud for Customer to the SAP CPQ quote.

### Knowledge Base Synchronization Translations

SAP CPQ allows administrators to synchronize translations in various languages which they are maintaining in Variant Configuration. Every time knowledge base synchronization is executed, translations are synchronized for all products and product versions obtained via that specific knowledge base synchronization. However, it is also possible to synchronize translations for products that were obtained via earlier synchronization jobs. Translations can be synchronized on the level of product names/descriptions, attribute names/descriptions, attribute value names/descriptions, and tabs. More information about synchronizing translations is available in Knowledge Base Synchronization Settings [page 84].

### Favorites

Variant Configuration product configurations can be added to Favorites from the Configurator. It is possible to save several configurations of the same product, either complete or incomplete, as separate favorites. Consequently, this functionality provides users with a practical workaround for material variants. Moreover, you cannot add products to favorites directly from the Catalog. Instead, you need to click Configure first, and then add them to favorites from the Configurator. A Variant Configuration item saved as a favorite permanently retains the original reference attributes, but the pricing attributes are updated every time the configuration is opened. For example, if a different quote is currently active and is providing a new context for executing the rules, the pricing attributes are updated according to the new context whenever the configuration is opened.

**i Note**

To be able to add incomplete configurations from favorites to the quote, make sure that Allow adding incomplete items is turned on for your user type in the Setup.

Note the difference between incomplete configurations (which lack an element that is required) and inconsistent configurations (in which there is a conflict). Inconsistent configurations can always be added to the quote, and are flagged as inconsistent in the quote.

### Conflicts

Additionally, the tab Conflicts displays in the responder whenever a conflict is created in the configurator (see the screenshot below). The tab lists all the conflicts occurring in the product along with their specific causes.
With the Variant Configuration integration, traffic is monitored via the SAP Passport, a payload/token attached to outbound API calls sent from SAP CPQ to SAP SAP Variant Configuration and Pricing and SAP Cloud Platform Integration.

**Note**

Product renewal is currently not supported for Variant Configuration products.

**Caution**

Items obtained from Variant Configuration via knowledge base synchronization only display in the catalog if they are flagged as root items. Consequently, configurable items which are not root items, and thus cannot be configured, do not appear in the catalog.

Only the items that are getting updated to a version from Variant Configuration can obtain the new flag. If the item version in Variant Configuration is the same as the item version in SAP CPQ, the flag is not obtained through knowledge base synchronization. If this is the case, and the problem is not solved through knowledge base synchronization, please contact the SAP CPQ Support team for assistance.

**Quote-First Approach for Variant Configuration Integration [page 95]**

When SAP CPQ is integrated with SAP Variant Configuration and Pricing, the modelling in SAP CPQ needs to be done in such a way that users land on a new quote, not the catalog.

**SAP Sales Area in SAP CPQ [page 96]**

SAP Sales Area is an eight-character code, which is a combination of sales organization, distribution channel and division.

### 1.4.4.1 Quote-First Approach for Variant Configuration Integration

When SAP CPQ is integrated with SAP Variant Configuration and Pricing, the modelling in SAP CPQ needs to be done in such a way that users land on a new quote, not the catalog.

Additionally, upon landing on a new quote, a Bill-to customer, which is synchronized from SAP ERP over the SAP Cloud Platform Integration to SAP CPQ, needs to be selected first. To make customer a required field that
needs to be filled out first, enable the When creating a new quote, require customer to be selected first toggle switch for a particular user type in Setup > Users > User Types.

**Effective Date**

Products synchronized from SAP Variant Configuration and Pricing can have different product versions. These product versions are obtained along with their own effectivity periods. Consequently, the Effective Date fields on the quote and the User Page are used to determine which product versions appear in the catalog. The effectivity period of the product version needs to match the Effective Date on the quote or User Page in order for that product version to appear in the Catalog. Additionally, the price displayed to the user depends on which product version is displayed.

Detailed information about the behavior of the Effective Date field is available on the Effective Date page.

**Related Information**

- Introduction to the Variant Configuration Integration [page 78]
- Technical Overview [page 91]
- Effective Date

### 1.4.4.2 SAP Sales Area in SAP CPQ

SAP Sales Area is an eight-character code, which is a combination of sales organization, distribution channel and division.

Before defining a sales area in SAP CPQ, you first need to:

- manually create markets in SAP CPQ
- manually create pricebooks for simple products in SAP CPQ, as prices for simple products will be sent from SAP ERP to SAP CPQ pricebooks via Upsert Pricebook With Distribution Chain UpsertPricebookWithDistributionChain method.

**Sales Organization**

Sales organization is market code in SAP CPQ. Markets related to SAP ERP sales organizations will have a market code that is equal to the sales organization code. Sales organization code is a four-digit alphanumeric, case-insensitive code.
Distribution Channel

Distribution channel is a field in SAP CPQ pricebooks. The field can be edited only when SAP CPQ is integrated with SAP ERP or S/4 HANA. Distribution channel is a two-digit alphanumeric, case-insensitive code.

Distribution chain is a combination of sales organization and distribution channel. The six-character code is also displayed in SAP CPQ pricebooks. The distribution chain code cannot be edited, as it is automatically created below the Distribution Channel field. Distribution chain is used for importing pricebook entries over the SAP CPQ API.
Division

A division code is a two-digit alphanumeric, case-insensitive code. Division is set up in SAP General Attribute Mappings [page 86].

Related Information

Define SAP General Attribute Mappings [page 86]
Introduction to the Variant Configuration Integration [page 78]

1.4.5 Pricing

You can select Variant Pricing or Pricebook Lookup as the pricing type for Variant Configuration products in SAP CPQ.

**Variant Pricing** is the type of pricing which can be assigned in Products and Products 2.0 in the Setup for products synchronized from SAP Variant Configuration and Pricing. When **Variant Pricing** is the selected pricing type, product prices are synchronized from SAP Variant Configuration and Pricing. Therefore, this type of pricing is always selected by default for synchronized configurable products.

Moreover, if **Variant Pricing** is selected in the Setup, synchronized simple products can also obtain prices from SAP Variant Configuration and Pricing. Alternatively, you can select Pricebook Lookup for simple products. In that case, prices for simple products are sent from SAP ERP to SAP Cloud Platform Integration over IDocs. SAP Cloud Platform Integration then calls SAP CPQ SOAP API for pricing and loads prices into SAP CPQ pricebooks (which is why pricebooks need to be manually created in SAP CPQ). When updated, prices in SAP ERP will be synchronized with pricebooks in SAP CPQ.

---

**Note**

When prices obtained from SAP Variant Configuration and Pricing are being used and the market has a defined market factor and currency rate, the obtained price is multiplied with the market factor and the currency rate.

Base price values for simple and configurable products are displayed in the Catalog if a **Pricing Procedure Name for Catalog Prices** is defined in SAP General Attribute Mappings [page 86] and maintained in SAP ERP [page 100]. Prices displayed for products synchronized from SAP Variant Configuration and Pricing in the Catalog do not include any preselections that were defined for that configurable product, but only the base price.

Prices in the Catalog are obtained from the pricing service by sending a batch request to api/v1/statelesspricing. The number of items in the batch request depends on the SAP CPQ Catalog paging (for example, if 50 products are rendered in the Catalog, the batch request is for 50 products).

Prices for configurable products are obtained in real time from SAP Variant Configuration and Pricing and displayed in the responder. The responder displays a complete breakdown of the pricing for products originating from SAP Variant Configuration and Pricing, which consists of the base price and the sum of surcharges. The breakdown can be modeled to retrieve any pricing function from the backend (via base price, selected options and variant conditions attached to the characteristics in SAP General Attribute Mappings).
Once a product is configured, the pricing breakdown is displayed in the Configuration Summary of the responder (see the screenshot). If the surcharges do not have descriptions, they are named after their keys.

You can write a script to access the pricing response from SAP Variant Configuration and Pricing. That way, you can expose a complete overview of the pricing procedure. Additionally, you can use scripting to assign the values of various condition types to item custom columns in the quote. By doing that, you can make the values of individual condition types visible on the quote user interface. More information is available in Map Pricing Response to Quote Columns [page 102].

**Note**

For the pricing service to work properly within this integration, make sure that the Number of Decimals in Calculations is set to a value no greater than 3 in Setup > Pricing/Calculations > Rounding Setup for Quantity (or for Default if Quantity is not defined here).

Set Up Pricing Procedure in SAP ERP [page 100]

In order to set up the pricing procedure, adjustments need to be maintained in SAP ERP, along with the settings in SAP General Attribute Mappings on the SAP CPQ side.

Map Pricing Response to Quote Columns [page 102]

The pricing response from SAP Variant Configuration and Pricing can be made visible in SAP CPQ if you map it to quote columns.

Pricing Behavior for Products Created/Edited via SOAP API [page 105]
The Variant Pricing mechanism behaves in a specific way in case products are created or edited via SOAP API.

### 1.4.5.1 Set Up Pricing Procedure in SAP ERP

In order to set up the pricing procedure, adjustments need to be maintained in SAP ERP, along with the settings in SAP General Attribute Mappings on the SAP CPQ side.

#### Prerequisites

In SAP ERP, you must activate business function `ERP_IPC_INT_PRC` (transaction `SFW5`) before you can assign condition functions to condition types. If you do not activate the business function, you can still see the total price (net value without shipping costs excluding tax) on the product configuration page, but the price is not broken down into the base price and the option surcharges.

**i Note**

If, in your system, the transaction `SFW5` does not list business function `ERP_IPC_INT_PRC`, please implement SAP Note 2125185 first. In SAP S/4HANA On-Premise, this is activated by default.

#### Procedure

1. Create condition functions for the condition types you use for the base price (net value) and the total of the price-relevant options selected, in Customizing for SAP ERP, under Sales and Distribution Basic Functions Pricing Pricing Control Define Condition Types.

   The graphics display the condition type PR00 Price with the condition function ZSS1 for the Base price, and the condition type VA00 Variant Price with the condition function ZSS2 for the Selected options. In step 3 of this procedure, you will enter these condition functions in SAP CPQ in the Base Price and Selected Options fields.
2. Use these condition types in the pricing procedure that you use for configurable products in *Customizing for SAP ERP* under **Sales and Distribution > Basic Functions > Pricing > Pricing Control > Define And Assign Pricing Procedures**

In the example, the pricing procedure with the name WECNUS is created for configurable products, and condition types PRO00 *Price* and VA00 *Variant Price* are used in it. In step 3 of this procedure, you will enter the procedure name in SAP CPQ in the field *Pricing Procedure Name* (in this example the value is WECNUS).
3. In SAP CPQ, in SAP General Attribute Mappings [page 86], fill out the fields Pricing Procedure Name, Base Price and Selected Options with the procedure name and condition functions from SAP ERP SAP ERP, and click Save.

The mapped pricing procedure is now in use.

**Related Information**

Define SAP General Attribute Mappings [page 86]
Technical Overview [page 91]
Introduction to the Variant Configuration Integration [page 78]

### 1.4.5.2 Map Pricing Response to Quote Columns

The pricing response from SAP Variant Configuration and Pricing can be made visible in SAP CPQ if you map it to quote columns.

A pricing procedure combines different types of charges (such as the list price, freight, discounts and surcharges) into a single procedure, where the sub-totals and the totals comprising the net amount are defined. In other terms, a pricing procedure defines a group of condition types in a particular sequence.

When SAP CPQ is integrated with SAP Variant Configuration and Pricing, pricing procedures are defined in SAP ERP and, from there, via SAP Variant Configuration and Pricing, replicated into SAP CPQ (as long as Variant Pricing is selected for products obtained via knowledge base synchronization). It is possible to map pricing procedures into SAP CPQ via the fields described in Define SAP General Attribute Mappings [page 86]. When a pricing procedure is mapped this way, the arriving response payload is stored in the SAP CPQ database, in the column `pricingResponseJson` of the table `sys_VC_Item_Pricing`. The pricing response payload contains information about the conditions from the pricing procedure.
A pricing procedure with the name ZRT5201 exists in SAP ERP, in which various condition types are defined. One of them is the condition type VA00 with the step number 10 and the condition value 40. The information about the condition type is stored in SAP CPQ as $conditionType$ (VA00), $conditionStep$ (10) and $conditionValue$ (40).

By default, the List Price column in the quote displays the sum of the base price and the variant conditions from the pricing procedure. There are multiple ways pricing can be set up here:

- SAP CPQ-specific discounts and surcharges can be added on top of the List Price. This is the out-of-the-box scenario.
  - SAP CPQ Quote Header Price Type → SAP S/4HANA/SAP ERP Sales Order Header Price Condition
    - $Additional Discount \cdot Discount Percent$ → RB00
    - $Additional Discount \cdot Discount Amount$ → HA00
  - SAP CPQ Quote Item Price Type → SAP S/4HANA/SAP ERP Sales Order Item Price Condition
    - List Price → PR00
    - Rolled Up Discount Percent → K007

- A pricing procedure breakdown can be mapped into multiple columns and the existing conditions (discounts, surcharges and so on) can be edited.
- A pricing procedure breakdown can be mapped into multiple columns, along with SAP CPQ-specific discounts and surcharges (which do not exist in the pricing procedure).

**Note**

When standard condition types are edited, they are sent to the order via the standard integration package. In case of condition types mapped through scripting, additional SAP Cloud Platform Integration content needs to be created for the edited conditions to be used. When the user changes the quantity or the configuration, the Reconfigure action is performed. The application parameter Calculate Auto-Reconfigure only items with modified ItemQuantity attribute determines whether the manual changes are maintained or lost.

You can add discounts modeled in SAP CPQ that are reflected as values of the specified condition types (RB00, HA00, K007) when you place an order from the Quote 1.0 engine in SAP CPQ. This is part of the standard package for the SAP CPQ – SAP ERP integration (see Value Mapping). Mapping other condition types is explained below and is done by accessing the pricing response from SAP Variant Configuration and Pricing and mapping the condition to a quote column.

**Note**

Variant Configuration integration is available for both the Quote 1.0 and Quote 2.0 engines. The integration works in the same way on both quote engines, the only difference being that in the integration for the Quote 2.0 engine, the Copy Quote and Place Order to ERP actions are currently not supported. Fixing these two limitations is on the team's roadmap and will be handled with priority in the upcoming releases.

Additionally, you can create promotions that apply discounts to standard fields mapped to these conditions. All SAP CPQ pricing elements can be added after the prices obtained from the pricing service. SAP CPQ pricing elements cannot be processed in the pricing service. However, you can add an SAP CPQ pricing element, map it to SAP ERP when placing the order, and remodel the pricing procedure in SAP CPQ to maintain correct values and calculations.

All the information from the pricing response from SAP Variant Configuration and Pricing is fully accessible through scripts. It is possible to make condition types from the pricing procedure visible and/or editable on the
user interface of the quote. To do that, you need to use a script to map individual condition types to custom item columns.

**Prerequisites**

Complete the following prerequisites before mapping the pricing response into quote columns.

1. Create a dedicated **quote item custom field** to which you want to map the condition type.
2. To display the field on the quote, you need to add it to the quote XSLT, which is defined per **user type**.
3. Assign a value to the quote item custom field’s key in the **dictionary**, to enter a name for the field.
4. Add the field to the **list of editable fields**.

**Procedure**

1. Go to | Setup | Develop | Global Scripts | Add New |.
2. Write a script that will map the pricing response breakdown to quote item custom fields.
3. Select an execution event for the script in the **Events** tab.
4. Click **Save**.

The value of the condition from the pricing response is now visible in the quote item custom field to which it was mapped via the script.

Here is an example script which can be used here:

```python
for mainItem in Quote.MainItems:
    if not mainItem.ParentItemGuid:  # checks for the root item, to get the entire pricing response
        x = 0
        y = 0
        for condition in mainItem.VCPricingPayload.Conditions:
            if condition.conditionType == 'PR00':  # checks for the certain condition type (In this case: List price)
                x += condition.ConditionValue  # summarizing values of condition type PR00
            elif condition.conditionType == 'VA00':
                y += condition.ConditionValue
        mainItem["BasePrice"].Value = x  # assigns condition values to CPQ Quote Item Custom Field
        mainItem["PricingSurcharge_1"].Value = y
```

**Making the Quote Item Custom Field Editable**

There are additional requirements which you need to complete to make the quote item custom field editable:

- Write a custom calculation for that field.
Once a condition type is mapped to a quote item custom field, SAP Variant Configuration and Pricing no longer provides the pricing information, so a custom calculation must be used in order for any edits to apply.

Adapt the standard IFlow for the *Place Order* action in SAP Cloud Platform Integration by manually mapping condition types and their respective values exposed in SAP CPQ columns to the corresponding elements from SAP ERP. For example, the quote item custom field *Surcharge 1* should be mapped to the corresponding condition from the SAP ERP pricing procedure *ZRt52t*. This is currently applicable only for Quote 1.0.

## Related Information

- Technical Overview [page 91]
- Define SAP General Attribute Mappings [page 86]
- Set Up Pricing Procedure in SAP ERP [page 100]
- Quote 2.0

### 1.4.5.3 Pricing Behavior for Products Created/Edited via SOAP API

The *Variant Pricing* mechanism behaves in a specific way in case products are created or edited via SOAP API.

- If the SAP Variant Configuration and Pricing integration is enabled and the *Pricing Mechanism* node is not listed in the *ImportMaterialsFromERP* SOAP API call, the resulting products have *Variant Pricing* selected as the pricing mechanism.
- If the SAP Variant Configuration and Pricing integration is not enabled and the *Pricing Mechanism* node is not listed in the *ImportMaterialsFromERP* SOAP API call, the resulting products have the default pricing mechanism selected.
- Variant Pricing can be selected only when the SAP Variant Configuration and Pricing integration is enabled, and the product is flagged as an SAP product (*IsSAPProduct*) = 1.

## Related Information

- Pricing [page 98]
- Technical Overview [page 91]
1.5 SAP Subscription Billing Integration

Integrating SAP CPQ with SAP Subscription Billing allows you to conduct subscription-based business, which is gaining prominence in today’s economy.

This integration uses SAP Cloud Platform Integration as middleware for communicating data across the systems.

Information about the API used in this integration is available in the following topics:

- Quote API
- Rate Plans API

Prerequisites [page 106]
These are the general prerequisites that need to be completed so that you can integrate SAP Subscription Billing with SAP CPQ.

Setup Adjustments on the SAP CPQ Side [page 107]
To integrate SAP CPQ with SAP Subscription Billing, you need to complete the procedures listed here.

Basic Integration Scenario [page 112]
This is a basic integration scenario for the SAP Subscription Billing integration.

Create Subscription Product [page 113]
A subscription product is a product which utilizes Subscription Pricing. In other words, a subscription product can have a base price (one-time charge), periodic and usage-based charges which are charged over the lifetime of the subscription. Subscription products must be simple products.

Set Values for Rate Plan Template Charges [page 115]
In Extended Rate Plan, you can define the values for the rate plan template charges, as well as create and define snapshots.

Use Subscription Products on User Side [page 116]
This topic explains how subscription products are used on the user side of SAP CPQ.

Quote Item Fields on Subscription Products [page 118]
When SAP CPQ is integrated with SAP Subscription Billing, the quote layout changes: quote item columns are replaced with separate cells on the item level, and subscription product details can be expanded or collapsed by clicking View/Hide Charges. Furthermore, additional standard fields appear on the quote for subscription products only.

1.5.1 Prerequisites

These are the general prerequisites that need to be completed so that you can integrate SAP Subscription Billing with SAP CPQ.

- Quote 2.0 should be enabled on your SAP CPQ tenant.
  It is important to note that this integration is not supported on Quote 1.0.
- SAP Cloud Platform Integration should be used as the middleware.
  The integration package containing integration flows and value mappings for this integration is available in the SAP API Business Hub. Information on setting up the integration flows and value mappings is
available in Integration of SAP Subscription Billing with SAP CPQ – SAP Cloud Platform Integration Content.

- Customer master data must be replicated from a back-end system (for example, SAP S/4HANA) to both SAP CPQ and SAP Subscription Billing. The external reference of the customer must be available both in SAP CPQ and SAP Subscription Billing so that the customer is uniquely identifiable.
- Complete the necessary adjustments [page 107] in the SAP CPQ Setup.

Related Information

Setup Adjustments on the SAP CPQ Side [page 107]
Basic Integration Scenario [page 112]

1.5.2 Setup Adjustments on the SAP CPQ Side

To integrate SAP CPQ with SAP Subscription Billing, you need to complete the procedures listed here.

1. Create markets which you will be using to set the values for rate plan charges and specify the market timezones.
2. Set up a connection [page 108] with SAP Cloud Platform Integration in Setup > Providers > Subscription to Events.
5. To be able to add subscription products to the quote, the checkbox Prevent users from quoting part numbers that haven’t got a price in pricebooks must be deselected in Setup > Pricebooks > General Settings.
6. To successfully integrate any system with SAP CPQ, the application parameter Allowed origins for the CORS filter needs to be configured in the way explained in Application Parameters.

i Note

Other integrations which use SAP Cloud Platform Integration require you to configure it in the Providers section. However, this does not apply to the SAP Subscription Billing integration, for which you need to set up the connection with SAP Cloud Platform Integration via Subscription to Events [page 201], instead.

If other integrations are enabled at the same time as the SAP Subscription Billing integration, and they also utilize SAP Cloud Platform Integration, SAP Cloud Platform Integration should be configured in two places at the same time:

- In Providers (for the other integrations).
- In Subscription to Events (for the SAP Subscription Billing Integration only).

The graphic illustrates where the elements of the integration are set up (in SAP Subscription Billing or in SAP CPQ).
Adding Subscriptions to Events [page 108]
To enable SAP CPQ to send information to SAP Subscription Billing over SAP Cloud Platform Integration, you need to create subscriptions to events, which send notifications from SAP CPQ to SAP Cloud Platform Integration whenever certain events are triggered.

Set Up SAP Subscription Billing [page 110]
Complete this procedure to set up the SAP Subscription Billing integration in the Setup.

Adjust Number of Decimals in Rounding Setup [page 111]
Complete these adjustments in the Rounding Setup in order for the integration to work properly.

1.5.2.1 Adding Subscriptions to Events
To enable SAP CPQ to send information to SAP Subscription Billing over SAP Cloud Platform Integration, you need to create subscriptions to events, which send notifications from SAP CPQ to SAP Cloud Platform Integration whenever certain events are triggered.

_i Note_
You can learn how to create subscriptions to events in Subscription to Events [page 201].

The SAP Cloud Platform Integration package provided for this integration contains the following IFlows:
- Replicate Subscription Products from CPQ to Subscription Billing
- Replicate Rate Plans from CPQ to Subscription Billing
- Replicate CPQ Quotes to Subscription Billing Provisioning Request

In Subscription to Events, you need to create subscriptions to events for each of the three IFlows. Listed below are sample subscriptions you can create for this integration.

→ Remember
The toggle switch Use for Subscription Billing products should be turned on for each of these events (except the QuoteProvisioned event).
Create Product

- **Name**: Create Product
- **Description**: this field is optional.
- **Webhook URL (required)**: enter the URL for the Replicate Subscription Products from CPQ to Subscription Billing IFlow provided in the SAP Cloud Platform Integration package.
- **Webhook HTTP method (required)**: POST
- **Event name (required)**: ProductCreated
- **Authentication type (required)**: Basic
- **Username (required)**: this should match the information from SAP Cloud Platform Integration.
- **Password (required)**: this should match the information from SAP Cloud Platform Integration.

Update Product

- **Name**: Update Product
- **Description**: this field is optional.
- **Webhook URL (required)**: enter the URL for the Replicate Subscription Products from CPQ to Subscription Billing IFlow provided in the SAP Cloud Platform Integration package.
- **Webhook HTTP method (required)**: POST
- **Event name (required)**: ProductUpdated
- **Authentication type (required)**: Basic
- **Username (required)**: this should match the information from SAP Cloud Platform Integration.
- **Password (required)**: this should match the information from SAP Cloud Platform Integration.

Create Rateplan

- **Name**: Create Rateplan
- **Description**: this field is optional.
- **Webhook URL (required)**: enter the URL for the Replicate Rate Plans from CPQ to Subscription Billing IFlow provided in the SAP Cloud Platform Integration package.
- **Webhook HTTP method (required)**: POST
- **Event name (required)**: ProductCreated
- **Authentication type (required)**: Basic
- **Username (required)**: this should match the information from SAP Cloud Platform Integration.
- **Password (required)**: this should match the information from SAP Cloud Platform Integration.

Update Rateplan

- **Name**: Update Rateplan
- **Description**: this field is optional.
- **Webhook URL (required)**: enter the URL for the Replicate Rate Plans from CPQ to Subscription Billing IFlow provided in the SAP Cloud Platform Integration package.
- **Webhook HTTP method (required)**: POST
- **Event name (required)**: ProductUpdated
- **Authentication type (required)**: Basic
- **Username (required)**: this should match the information from SAP Cloud Platform Integration.
- **Password (required)**: this should match the information from SAP Cloud Platform Integration.
Quote Provisioned

- **Name**: Quote Provisioned
- **Description**: this field is optional.
- **Webhook URL (required)**: enter the URL for the Replicate CPQ Quotes to Subscription Billing Provisioning Request IFlow provided in the SAP Cloud Platform Integration package.
- **Webhook HTTP method (required)**: POST
- **Event name (required)**: Quote Provisioned
- **Authentication type (required)**: Basic
- **Username (required)**: this should match the information from SAP Cloud Platform Integration.
- **Password (required)**: this should match the information from SAP Cloud Platform Integration.

Related Information

- Setup Adjustments on the SAP CPQ Side [page 107]
- Subscription to Events [page 201]

1.5.2.2 Set Up SAP Subscription Billing

Complete this procedure to set up the SAP Subscription Billing integration in the Setup.

Context

All the information needed in this section should be provided by the administrator managing SAP Subscription Billing. Settings are configured on the tenant level.

Procedure

1. Go to Setup Providers SAP.
   - Available SAP providers display.
2. Click SAP Subscription Billing.
   - A new page displays.
3. Turn on the Enable SAP Subscription Billing Integration toggle switch.
4. In Authentication Settings, specify the following:
   - **Authentication base URL** – the URL of the SAP Subscription Billing authorization service. The URL should be ending with /oauth/token.
   - **Client ID** – OAuth 2.0 client identifier issued by the SAP Subscription Billing authorization provider.
5. In **SAP Subscription Billing Base URL and Endpoints**, specify the following:
   - **Base URL** – the root SAP Subscription Billing API URL.
   - **RatePlanTemplate Endpoint** – relative endpoint to the API that handles rate plan templates.

6. Click **Save**.

   The settings are now configured.

---

**Related Information**

Prerequisites [page 106]
Basic Integration Scenario [page 112]

---

### 1.5.2.3 Adjust Number of Decimals in Rounding Setup

Complete these adjustments in the **Rounding Setup** in order for the integration to work properly.

#### Procedure

1. Go to **Setup > Pricing/Calculations > Rounding Setup**.
2. Click 🖌 next to the **Default** column name.
   
   The editing page opens.
3. In **Number of Decimals in Calculations**, enter **4**.
4. In **Number of Decimals when Rounding**, enter **4**.
5. Click **Save**.

   The integration is fully set up.

   The **Default** values you specify here apply to all calculations and rounding whenever specific values are not provided. If other values are specified in the **Rounding Setup**, change them all to **4** for both **Number of Decimals in Calculations** and **Number of Decimals when Rounding** by repeating this procedure for every row in the table.

---

⚠️ **Caution**

The number of decimals specified in the **Rounding Setup** applies to values throughout SAP CPQ. It is not limited to the features introduced with the SAP Subscription Billing integration.
1.5.3 Basic Integration Scenario

This is a basic integration scenario for the SAP Subscription Billing integration.

Administrator Side

First, you need to create a subscription product [page 113] in SAP CPQ.

You create a subscription product by setting the pricing type Subscription Pricing for a product in Setup > Products 2.0 (using Products is not supported). When you select Subscription Pricing, you can define the contract length and the minimum term (which can be edited on the quote afterwards). Afterwards, choose a rate plan template and a market and set the values for the charges [page 115] of the rate plan template. Rate plan templates that have been published in SAP Subscription Billing are automatically retrieved when the integration is enabled. Additionally, it is important to create a market beforehand. The market should have a defined timezone, and SAP CPQ markets should match the markets in SAP Subscription Billing. Only one rate plan template can be selected per product, and one rate plan corresponds to one market.

Additionally, you can create snapshots [page 115], which are dates from which specific charge values within a rate plan become valid. A rate plan template with defined charge values becomes a rate plan, and the product with the rate plan is replicated into SAP Subscription Billing once it is saved in SAP CPQ.
User Side

After the subscription product is created in the Setup, the user needs to select the subscription product in the SAP CPQ catalog and add it to the quote [page 116]. All the details of the rate plan are visible in the quote when a subscription product is added. Additionally, they are also visible in the Catalog if you click Pricing Details.

Fill in the pricing details in the quote. You can generate a document displaying elements of the subscription plan by using the dedicated template tags. Once you are satisfied with the quote, click Place Order. This triggers the replication of the quote to SAP Subscription Billing, resulting in the creation of a provisioning request. Once orchestrated, this procedure results in the creation of subscriptions, and, ultimately, bills.

Related Information

SAP Subscription Billing Integration Tags
Calculations for SAP Subscription Billing Integration
Limitations and Known Issues 2008

1.5.4 Create Subscription Product

A subscription product is a product which utilizes Subscription Pricing. In other words, a subscription product can have a base price (one-time charge), periodic and usage-based charges which are charged over the lifetime of the subscription. Subscription products must be simple products.

Prerequisites

SAP CPQ must be integrated with SAP Subscription Billing. Rate plan templates must be created and published in SAP Subscription Billing.

Procedure

1. Go to Setup > Product Catalog > Products 2.0.
2. Click Add New Product.
3. Fill out the Definition tab with relevant information.
   The name of the product should only contain letters (a-z, A-Z), numbers (0-9) and the underscore (_), slash (\) and hyphen (-) symbols.
4. In the Pricing tab, select Subscription Pricing as the Pricing Type.
   New fields display.
5. Fill out or make a selection in the three fields:
   ○ *Contract Length* – specify the number of months.
     This field determines the total duration of the subscription. This field can be edited later in the quote.
   ○ *Minimum Term* – specify the number of months.
     This field determines the shortest period after which the subscription can be cancelled (that is, the shortest period after which the contract can be terminated). The minimum term cannot be longer than the contract length. However, the minimum term can be greater than 0 if the contract length value is 0 (an unlimited subscription can still have a defined minimum term). This field can be edited later in the quote.
   ○ *Rate plan templates* – click this button to select a rate plan template.
     The rate plan templates in the list are automatically retrieved from SAP Subscription Billing when the integration is enabled, and the rate plan templates have been published.

6. Click *Select*.
   You have now selected a rate plan template. The *Market* dropdown menu appears.

7. Select a market in the *Market* dropdown menu.
   When you assign a market to a subscription product, the prices for the product appear only when that market is selected. Only one market can be selected per rate plan. Moreover, the timezone which you specify for the market in *Markets* is applied to the start and end dates and the duration of the subscription product. These fields are defined as midnight of the market’s timezone.

8. After selecting the market, click *Add*.
   A table appears, listing the market and currency that are specified for the rate plan.

9. Click *Rate Plan* to open the extended rate plan [page 115], where you can set the values for the charges of the product, which will apply for the market in the corresponding row.
   A rate plan template becomes a rate plan once you specify the charge values for it. You can select one rate plan template per product, but you can create multiple rate plans which will apply for different markets.

   Alternatively, you can delete rows in the table by clicking 

10. Once you have defined the charges for the rate plan, click *Save*.
    The subscription product is now added to the *Catalog*. After clicking *Save*, it is not possible to change the pricing type or remove selected rate plans (however, you can add new rate plans or edit existing ones).

   **Note**
   You can select and add multiple markets (and, consequently, define multiple rate plans) for a single product. They will appear as new rows in the table. You need to set the charges for each specific market by clicking *Rate Plan* in the corresponding row and setting the parameter values.

**Related Information**

- Set Values for Rate Plan Template Charges [page 115]
- Use Subscription Products on User Side [page 116]
- Create Snapshot [page 115]
1.5.5 Set Values for Rate Plan Template Charges

In Extended Rate Plan, you can define the values for the rate plan template charges, as well as create and define snapshots.

Procedure

1. In Parameters, define the Initial Value for every parameter.

   Parameters are predefined as part of the rate plan template in SAP Subscription Billing. The currency for parameter values is defined under Currency in the Extended Rate Plan section.

2. (Optional) In the Charges section, you can see a detailed overview of the charges, updated with the specified parameter values.

   It is possible to set the fixed amount for a charge without using a parameter, but that can only be done on the SAP Subscription Billing side.

3. If all the steps from Create Subscription Product [page 113] are complete, click Save to create the subscription product.

   The product is now added to the catalog.

i Note

   SAP Subscription Billing documentation provides detailed information about the following elements:
   ○ Rate plans
   ○ Rate plan templates
   ○ Charge types and pricing models

Create Snapshot [page 115]

   A snapshot allows you to set charges that are only valid from a specified date within the total duration of the subscription term. You can create multiple snapshots for the same product. You create a snapshot by selecting a starting date, and it is valid from that date until the starting date of the next snapshot (or indefinitely, if no such date is defined).

1.5.5.1 Create Snapshot

   A snapshot allows you to set charges that are only valid from a specified date within the total duration of the subscription term. You can create multiple snapshots for the same product. You create a snapshot by selecting a starting date, and it is valid from that date until the starting date of the next snapshot (or indefinitely, if no such date is defined).
### Procedure

1. Go to Setup ➔ Product Catalog ➔ Products 2.0 ➔ Add New Product ➔ Pricing ➔ Extended Rate Plan.
2. In the Add Snapshot field, select the starting date for the snapshot.
   - The starting date is when the snapshot prices begin to apply. To use a snapshot, you need to set an effective date encompassed by that snapshot for the subscription product in the quote.
3. Click Add.
   - A snapshot column is added to the Parameters table, next to the Initial Value column and any previous snapshots. The heading of the column is the effective date you selected in the previous step.
4. Fill out the parameter values in the snapshot column.
5. (Optional) In the Charges section, select the effective date of the snapshot under Show.
   - A detailed overview of the charges with the snapshot values is displayed.
6. Click Save.
   - The snapshot of the rate plan is now saved for the selected subscription product. If the effective date of this product on the quote matches the period when the snapshot is active (starting with the snapshot initial date), the prices defined for the snapshot are displayed, instead of the prices defined in the Initial Value column of the rate plan.

### Related Information

- Create Subscription Product [page 113]
- Set Values for Rate Plan Template Charges [page 115]

### 1.5.6 Use Subscription Products on User Side

This topic explains how subscription products are used on the user side of SAP CPQ.

### Prerequisites

This integration uses the quote-first approach, which means that subscription products can only be added to an existing quote. Before starting the procedure below, create a new quote and select the market in which you previously defined the values for the rate plan charges.

### Context

Once you finish creating a subscription product [page 113], it appears in the Catalog with other products. To see the prices for the subscription product in the catalog, the market for which the rate plan is defined must be
selected as the default market on the User Page. If the prices are missing for a subscription product, you will not be able to add it to the quote. In addition, by clicking on the subscription product in the Catalog, you can see product-related information in the Product Details and rate plan information in the Pricing Details tab.

**Note**

After adding a subscription product to the quote, you cannot select a market on the quote which does not correspond with a defined rate plan for the added product. Instead of the market changing, an error message displays, and upon changing the tab, closing or saving the quote, the market resets to the previously selected market.

**Procedure**

1. Find your product in the Catalog and click Add to add it to the quote.
   
   You don’t need to configure the product, since subscription products are simple products.

2. Open the quote.

3. Locate the subscription product under Products.
   
   A subscription product contains only the standard fields which are specific to the SAP Subscription Billing integration. See Quote Item Fields on Subscription Products [page 118].

4. (Optional) Click View charges.
   
   A complete breakdown of the charges is displayed.

5. (Optional) To set discounts for the charges, click on the editable discount value and enter the new value.
   
   The prices in the quote are now updated with the new discount values.

6. (Optional) Specify the remaining values for the product.
   
   The editability of the fields can be changed in Setup Quote Quote Fields and Calculations.

7. (Required) Fill in the customer details.

8. (Optional) Click Generate Documents to begin the document generation procedure.
   
   The document you generate will contain all the information about the subscription product and its charges.

9. Click Place Order to finalize your order.
   
   All the information from the quote is now sent to SAP Subscription Billing, where a provisioning request is created.

**Note**

If you want your document generation template to retrieve information related to subscription products, you need to use the dedicated tags in the template.
1.5.7 Quote Item Fields on Subscription Products

When SAP CPQ is integrated with SAP Subscription Billing, the quote layout changes: quote item columns are replaced with separate cells on the item level, and subscription product details can be expanded or collapsed by clicking View/Hide Charges. Furthermore, additional standard fields appear on the quote for subscription products only.

The standard fields specific to the integration are the following:

- **Effective Date**
- **Contract Start Date**
- **Contract Length**
- **Minimum Term**
- **Contract End Date**
- **Minimum Term End Date**

The following CTX tags can be used for retrieving information from item fields on subscription products:

- `<CTX( Quote.CurrentItem.EffectiveDate )*>`
- `<CTX( Quote.CurrentItem.ContractStartDate )*>`
- `<CTX( Quote.CurrentItem.ContractLength )*>`
- `<CTX( Quote.CurrentItem.MinimumTerm )*>`
- `<CTX( Quote.CurrentItem.ContractEndDate )*>`
- `<CTX( Quote.CurrentItem.MinimumTermEndDate )*>`

**i Note**

It is possible for the effective date specified for the product to be an earlier date than the contract start date. That way, a previously used promotion can be reused for a new contract. You can learn about the product effective date in detail on the **Effective Date** page.

These fields are editable by default (the end date values are calculated automatically when you specify the length and the start date). They only appear on subscription products, while non-subscription products contain the same fields as they did before the integration was enabled.

Along with the standard fields, **standard calculations for subscription products** are introduced in the scope of the integration.

**i Note**

Quote item custom fields do not display on subscription products.
1.6 SAP Cloud for Customer Integration

The following pages describe both integrations and contain step by step instructions on how to integrate SAP Cloud for Customer and SAP CPQ.

SAP Cloud for Customer - SAP CPQ integration is developed in the 1905 SAP CPQ release. Before that, users could use the legacy integration system that is obsolete as of the SAP CPQ 1905 release. Users who have the legacy integration set up when 1905 SAP CPQ is released will continue using it uninterruptedly after this release. The only difference that administrators will notice is that (Obsolete) is appended to the name of the tab C4C CRM Settings in Setup.

SAP Cloud for Customer - SAP CPQ Integration [page 119]
SAP CPQ can be integrated with SAP Cloud for Customer providing a streamlined workflow for users managing their sales quotes in SAP Cloud for Customer.

Legacy SAP Cloud for Customer - SAP CPQ Integration [page 177]
This section describes the legacy SAP Cloud for Customer - SAP CPQ integration that was available to users who set up the integration before the 1905 SAP CPQ release. This integration became obsolete in 1905 for new users, although old users can use it uninterruptedly.

1.6.1 SAP Cloud for Customer - SAP CPQ Integration

SAP CPQ can be integrated with SAP Cloud for Customer providing a streamlined workflow for users managing their sales quotes in SAP Cloud for Customer.

The new integration is based on the initially developed integration that was in use before the 1905 release, but it is improved in terms of the user interface, user experience and the overall performance of the integrated environments.

The main improvement is that the SAP CPQ quote now acts as the only quote in SAP Cloud for Customer, while the SAP Cloud for Customer sales quote is created in the background. After landing the first time to SAP CPQ, the sales quote created in SAP Cloud for Customer is no longer editable and opening it opens the quote in SAP CPQ. Additionally, the user journey on the SAP Cloud for Customer side starts on the sales quote and not on the opportunity, as in the old integration.

Users that want to integrate SAP Cloud for Customer and SAP CPQ after the 1905 release of SAP CPQ will not be able to choose between the two integrations, only the new one will be available.

The following pages describe the procedure to configure the Basic Integration between SAP Cloud for Customer and SAP CPQ using SAP Cloud Platform Integration (CPI). The instructions are focused on the users
who will be managing the integration in a new integration tenant, not in an existing tenant. Users with an
existing tenant will also find the instructions useful, although they may come across errors that are bound to
the existing modeling on their tenants.

Network Security team takes the responsibility of preparing the network environment across different systems
and its security aspects.

**Note**

If you have any queries or feedback on this documentation, please create a ticket for the CSN component
SV-RDS-CFC.

Configurable products synchronized from [SAP Variant Configuration and Pricing](page 78) to SAP Cloud for
Customer can be used in a scenario for the SAP CPQ integration.

- **General Prerequisites**
  - There are some general prerequisites that need to be fulfilled so that users could integrate SAP Cloud
    for Customer and SAP CPQ.
- **Configuration**
- **Manual Build Steps**
- **User Journey from SAP Cloud for Customer to SAP CPQ**

This topic describes a typical user journey for this integration.

### 1.6.1.1 General Prerequisites

There are some general prerequisites that need to be fulfilled so that users could integrate SAP Cloud for
Customer and SAP CPQ.

- SAP CPQ and SAP Cloud for Customer need to be synchronized in terms of product (material) master,
  markets and pricebooks via the Enterprise Resource Planning software (SAP ERP) or SAP S/4HANA. Here,
  we will use SAP ERP throughout examples.
- To have the [SAP ERP Integration](page 7) set up.
- **Configuration in SAP CPQ**
- To have an integration tenant in SAP CPQ. SAP Cloud for Customer and in SAP Cloud Platform Integration.

1. **SAP CPQ instance**
   - Ensure that your SAP CPQ instance is in place before you start implementing SAP CPQ integration
     with SAP Cloud for Customer.

2. **SAP Cloud for Customer tenant**
   - Ensure that you have your SAP Cloud for Customer tenant and Initial User details, these details will be
     provided as a provisioning mail by SAP.
   - **URL Example:** [https://<YourTenant>crm.ondemand.com](https://<YourTenant>crm.ondemand.com)

3. **SAP Cloud Platform Integration tenant**
   - The SAP Cloud Platform Integration (CPI) Runtime URL is included in the SAP CPI provisioning mail.
   - **URL Example:** [https://<YourTenant>-iflmap.hana.ondemand.com](https://<YourTenant>-iflmap.hana.ondemand.com)

- A matching user must be created in both systems.
- **Configuration in SAP Cloud for Customer**
- **Configuration in SAP Cloud Platform Integration**
To complete these requirements, you will need to contact different teams in SAP (for example, the SAP Cloud Platform Integration team must provide the URL and the endpoints that you need to enter in the Providers page in SAP CPQ).

Related Information

System Requirements for SAP Integrations [page 5]
SAP ERP Integration [page 7]
User Journey from SAP Cloud for Customer to SAP CPQ [page 155]

1.6.1.2 Configuration

The following sections describes all the settings required for this scope item. These can be divided into below groups:

Configuration in SAP CPQ [page 121]
   This section describes how to configure SAP CPQ for the purpose of integration.

Configuration in SAP Cloud for Customer [page 130]
   This section describes ways to configure SAP Cloud for Customer for the purpose of integration.

Configuration in SAP Cloud Platform Integration [page 147]
   This section describes the steps for configuring the integration of SAP CPQ with SAP Cloud for Customer using SAP Cloud Platform Integration (CPI).

Data Loading [page 153]

1.6.1.2.1 Configuration in SAP CPQ

This section describes how to configure SAP CPQ for the purpose of integration.

1.6.1.2.1.1 Configure Single Sign-On with Federation Settings

Purpose

With Single Sign-On enabled for both SAP Cloud for Customer and SAP CPQ system using a common identity provider, business user can access the target SAP CPQ system without a need to enter the login credentials.
Pre-requisite

You should have a license of SAP Cloud Platform Identity Authentication Services.

Procedure

1. Login to your SAP Cloud Identity Provider account with your admin user.
2. From the Left Panel, Navigate to Application and Resources Tenant Settings.
3. Choose SAML 2.0 Configuration.
4. On the new screen, Click on Download Metadata File.

**i Note**

This Metadata is needed when you setup the trust in your SAP Cloud for Customer.

5. Login to your SAP CPQ system using an Administrator User.
6. Click on SETUP in menu bar and choose Setup. Select Providers in left navigation bar and Select Federation Settings.
7. Click on Download metadata document button to download metadata file to your local system.

**i Note**

This Metadata file is needed when you setup the trust in your SAP Cloud Identity provider.

8. Click on Add new identity provider button.
9. Maintain the Name as CPQ-C4C.
10. Click on Upload metadata document and select the metadata file downloaded from SAP Cloud Identity Provider.
11. All the fields will be auto populated by metadata file.
12. Choose Name Id in Read UID From.
13. Click on Edit button under Routings.

14. Click on Add button, Switch On the Enabled field.
15. Name and URL are auto populated. Select Sign on relay state as your SAP CPQ system Domain.
16. Click on Save in Routings Panel.
17. Then click on Save Button in Federation Settings.
18. Click Deploy button at the end of federation settings screen. The federation settings take 5 minutes to deploy.
1.6.1.2.1.2 Define Provider settings

Procedure

1. Login to your SAP CPQ system using an Administrator User.
2. Click on SETUP in menu bar and choose Setup.
3. Select Providers in left navigation bar and then select SAP.
5. In Common Settings tab, Switch on Enable Integration.
6. Select BasicAuth in Authentication Mode dropdown.
7. Maintain Integration Username and Password as the username and password of Cloud Platform Integration User.
8. Under General Setting, Switch on Send customers along with quote payload.
9. Enter the Cloud Platform URL in REST API base URL address and a postfix /http to connect to http adapter.
○ **REST API base URL address**: the base REST API URL address of the SAP Cloud Platform Integration – SAP CPQ specific package. The value is delivered by the administrator and should not be changed on the fly. In addition, the value identifies a specific SAP Cloud Platform Integration environment, that is, the tenant.

○ **Endpoint that gets SAP object data details**: a relative endpoint address of the SAP Cloud Platform Integration artifact that gets an object from the SAP system over SAP Cloud Platform Integration. The call acts as a proxy to SAP OData services and can query any publicly visible object from SAP Cloud for Customer. Leave this field empty.

○ **Endpoint that processes posted data in xml format**: a relative endpoint of the SAP Cloud Platform Integration artifact that processes the posted data in the XML format. Leave this field empty.

○ **Endpoint that sends CPQ quote to SAP systems**: a relative endpoint of the SAP Cloud Platform Integration artifact that sends the SAP CPQ quote to SAP systems. Leave this field empty.

10. Click **Save**.
11. Navigate to **C4C CRM Settings** tab.
12. Switch on the button for **C4C is Connected**.
13. Note the Landing URL, this is the SAP CPQ SSO URL which will be used in Identity provider and SAP Cloud for Customer.
14. Maintain the below Endpoints:

○ **Endpoint that gets sales quote details in case when landing and creating a new quote** - a relative endpoint address of the SAP Cloud Platform Integration artifact that retrieves sales quote header details and customer data (bill-to and ship-to) in XML format. The artifact is called upon landing from SAP Cloud for Customer and creating a new quote in SAP CPQ. Provided by SAP Cloud Platform Integration.

○ **Endpoint that updates SAP sales quote from CPQ quote data** - a relative endpoint address of the SAP Cloud Platform Integration artifact that processes the quote that SAP CPQ sends in XML format. In addition, the endpoint updates a remote SAP Cloud for CustomerSAP Cloud for Customer sales quote that is linked to the SAP CPQ quote. Note that the sales quote is linked to the SAP CPQ quote after landing on SAP CPQ from the SAP Cloud for Customer sales quote. Provided by SAP Cloud Platform Integration.

15. **Automatically update sales quote** - if enabled, the sales quote in SAP Cloud for Customer will be automatically updated after clicking the **Save** action on the quote in each tab that contains quote custom fields and also after changing the status of the quote. There is also the **Update Sales Quote** action available in the workflow if users want to update the sales quote manually.

16. Click **Save**.
### 1.6.1.2.1.3 Configuring SAP Cloud Identity Provider

**Procedure**

1. Login to your SAP Cloud Identity Provider account with your admin user.
2. Click on **Applications** tile to create your SAP CPQ application.
3. Click on **Add** and enter your SAP CPQ system name. For Example: CPQ-C4C
4. Click **Save**.
5. In the new Screen, Click on Home URL and enter your CPQ SSO URL. For Ex: `http://<your_tenant>webcomcpq.com/fed/sapcpq/sapcrm/saplogin.aspx`
6. Click **Save**.
7. In the Trust, Choose SAML 2.0 Configuration and Import the CPQ metadata file you have downloaded earlier.
8. Click **Save**.
9. Configure Name ID Attribute as **E-Mail**.
10. Click **Save**.
11. Configure **Default Name ID Format** as **Unspecified**.
12. Click **Save**.

### 1.6.1.2.1.4 Create CPQ Communication User

**Purpose**

A Communication user need to be created in SAP CPQ System, which will be used in Cloud Platform Integration for basic authentication to CPQ.

**Procedure**

1. Login to your SAP CPQ system using an Administrator User.
2. Click on **SETUP** in menu bar and choose **Setup**.
3. In left navigation bar, select **Users** under **Users**.
4. Click on **Add New User** Button.
5. Enter the **First Name**, **Last Name** and **Email Address** for Communication User.
6. Enter **Username** as **COM_USER_CPQ** and maintain the **Initial Password**.
7. Choose User Type as Sales.
8. Select Administrator.
9. Select Company as CallidusCloud (CALD).
10. Click Save.

**i Note**
Login to CPQ system using the above user and reset the password and use same password to be used in CPI at the time of Creating User Credentials.

### 1.6.1.2.1.5 Pricing/Discounting Setup

**Create Markets**

**Purpose**
A market is created in SAP CPQ against every Sales Organization of SAP Cloud for Customer system. This will help in pricing determination in Sales Quote in CPQ system.

**Procedure**
1. Login to your SAP CPQ system using an Administrator User.
2. Click on SETUP in menu bar and choose Setup.
3. In left navigation bar, select Markets under Pricing/Calculations.
4. Click on Add New Button.
5. Enter the Market Code field as a Sales Organization of S/4HANA system.
6. Enter Marketing Name, Currency and Market Factor.

<table>
<thead>
<tr>
<th>Market Code</th>
<th>Market Name</th>
<th>Currency</th>
<th>Market Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1710</td>
<td>Sales Org US</td>
<td>USD</td>
<td>1</td>
</tr>
<tr>
<td>1010</td>
<td>Sales Org DE</td>
<td>EUR</td>
<td>1</td>
</tr>
</tbody>
</table>
7. Click **Save**.
8. In left navigation bar, select **Market Visibility** under **Pricing/Calculations**.
9. Click on **Add New** Button.
10. Select **Market Name** created in above step in Step 2 Section.
11. In Step 3 section enter **Visibility Condition** as 1.
12. Click **Save**.

### Create Pricebooks

**Purpose**

Pricing in the Sales Quote is determined based on market, Pricebook and product in CPQ system. Pricebook comprises of Distribution Channel and Product entries with their respective prices.

**Procedure**

1. In left navigation bar, select **Pricebooks** under **Pricing/Calculations**.
2. Under **Pricebooks** tab, search for the **Market Name** created in above steps.
3. Click on **Add New Pricebook** button.
   Enter the **Pricebook name**.
4. Enter the **Distribution Channel** similar to the distribution Channel of S/4HANA system against the Sales Organization.
   Example: 10
   This field is editable only when SAP CPQ is integrated with SAP ERP or S/4 HANA.
5. Check the **Visible to everyone** checkbox.
6. Click Save.
7. Pricing for the products in Entries tab should be created by Pricing Replication from your ERP system.

**Discounting**

**Purpose**

One of the major functionalities of SAP CPQ is on the fly discounts based on products, product types, product categories, users and user types. You can setup discounting in Sales Quote in your SAP CPQ system following below procedure.

**Procedure**

1. In left navigation bar, select Discounts/Multipliers under Pricing/Calculations.
2. Click on Add New button.
3. Click on Select button to maintain USER TYPE as Sales in Step 1 section.

⚠️ Note

The above selection will allow users with user type ‘Sales’ to give discounts in Sales Quote which we will be defining in next step. You can create any condition for discount selection with User, User Type, Product, Product Type or Category.

4. In Step 2 section, maintain multiplier as 1 for Minimal Value, Default Value and Maximum Value.
5. Maintain Maximum Value for Discount as 10. It will allow your Sales User to give maximum of 10 percent discount. Above 10 percent will be subject to approval.
6. In Step 3 section, maintain Condition as 1. You can create conditions using scripting language also.
7. Click Save.

Discounting Priorities

Purpose
In SAP CPQ you can provide discount to customer in Sales Quote based on User, User Type, Product, Product Type or Product Category conditions. There might come a case with two or more valid conditions. In such situation, which condition will take priority based on below procedure.

Procedure
1. In left navigation bar, select Discounting Priorities under Pricing/Calculations.
2. As we want Product to have highest Priority. Select Rank as 1.
3. Further Select Rank for Product Type as 2, Category as 3, User as 4 and User Type as 5.
4. Click Save.

1.6.1.2.1.6 Specific SAP CPQ Prerequisites

You can integrate SAP Cloud for Customer and SAP CPQ either in an existing SAP CPQ tenant or in a new one that is designed specifically for the integration.

The following adjustments are made by default in the new integration tenant and should be made manually if you use an existing tenant.

- Quote custom fields Opportunity Name and Opportunity ID are removed from the tenant. If you are setting up the integration in an existing tenant, please create a ticket for the SAP CPQ Support Team to disable these actions on that tenant.
• Actions Place Order, Copy, Reassign and Delete are not supported.

• Workflow:
  ○ action Place Order in ERP changes the status Open to Order Confirmation Pending.
  ○ action Order Status Update changes the status Order Confirmation Pending to Order Placed.

• No actions are available for customer roles.

• The end-user role is not available (the application parameter Use End User Role is set to NO). The integration supports only the bill-to and ship-to customer roles.

• Two custom fields are added: Requested Date and PO Number.

• The calculation type for the Valid to (previously Quote Expiration Date) custom field is changed to Once, when quote is created.

• The application parameter URLs that may Embed CPQ is populated with: *dev.sapbydesign.com

• Additional discount is removed from quotes [page 158] and from editable groups in Cart Fields Administration.

• The actions available for the old integration are removed: Update SAP Opportunity, Update Opportunity And Create Update SAP Sales Quote, Create Update SAP Sales Quote, Attach Document To SAP Opportunity, Attach Document To SAP Opportunity and Sales Quote, Attach Document To SAP Sales Quote.

• To successfully integrate any system with SAP CPQ, the application parameter Allowed origins for the CORS filter needs to be configured in the way explained in Application Parameters.

Related Information

Field Mapping [page 158]
General Prerequisites [page 120]
User Journey from SAP Cloud for Customer to SAP CPQ [page 155]

1.6.1.2.2 Configuration in SAP Cloud for Customer

This section describes ways to configure SAP Cloud for Customer for the purpose of integration.

Scoping [page 131]
This section describes the scope of the integration between SAP Cloud for Customer and SAP CPQ.

Fine Tuning [page 138]
This section describes the need to perform fine-tuning to support specific business requirements.

Configure Single Sign-On in SAP Cloud for Customer [page 145]
This section describes how to access the target SAP CPQ system without a need to enter the login credentials.
1.6.1.2.2.1 Scoping

This section describes the scope of the integration between SAP Cloud for Customer and SAP CPQ.

To complete the scoping, it is required to follow the steps below:

- First Log in with Initial User [page 131]
- Create Administrator with Initial User [page 131]
- Lock Initial User with the Newly Created Administrator [page 134]
- Scoping with Admin User [page 135]

1.6.1.2.2.1.1 First Log in with Initial User

1. Open the Frontend Access tenant URL in Web browser.
   Example: https://<host_name>.crm.ondemand.com
2. Input the initial user and password.
3. Select your log-on language, by default it is English.
4. Choose Sign In; change the password in the dialog box.
5. Choose OK.
6. Choose Yes.

1.6.1.2.2.1.2 Create Administrator with Initial User

1. Navigate to the below work center.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>BUSINESS CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>IMPLEMENTATION PROJECTS</td>
</tr>
</tbody>
</table>

2. Select the title First Implementation project, and then choose Open Activity List.
3. On the Prepare tab, choose Define Administrators for Project Team.
4. Choose Create Service Agents, maintain the following values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>ADMIN</td>
</tr>
<tr>
<td>Last Name</td>
<td>PROJECT</td>
</tr>
<tr>
<td>Phone</td>
<td>&lt;The Project Administrator or Contact’s Phone&gt;</td>
</tr>
<tr>
<td>E-Mail</td>
<td>&lt;The Project Administrator or Contact’s E-Mail&gt;</td>
</tr>
</tbody>
</table>

5. Click **Save** and choose Request User.

6. Click **Close**.

7. Close these opened pages.

8. Now sign out and then sign in the system again with project initial user.

9. Go to the below work center and choose **Go**.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMINISTRATOR</td>
<td>GENERAL SETTINGS</td>
</tr>
</tbody>
</table>

10. Select **Business Users** under the section **Users**.

11. Select the **User ID**.

   Example: PROJECTADMIN7000000

12. Go to **Edit Attributes** Attributes, maintain the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td>&lt;USERID&gt; For example: PROJECTADMIN</td>
</tr>
<tr>
<td>Password</td>
<td>&lt;Password&gt;</td>
</tr>
</tbody>
</table>

13. Click **Save** then choose **Edit Access Rights**, choose **Yes** to save the attributes change.
14. In the **EDIT ACCESS RIGHTS** screen, select tab **WORK CENTER AND VIEW ASSIGNMENT** and make below assignment by selecting the checkbox.

15. Click **Save** twice.
16. Click Sign Out on the top right corner and choose Yes.

1.6.1.2.2.1.3  Lock Initial User with the Newly Created Administrator

1. Choose Restart Application or sign in again.
2. Input the administrator user and password created and choose Sign In.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td><code>&lt;USER ID&gt;</code> For example: PROJECTADMIN</td>
</tr>
<tr>
<td>Password</td>
<td><code>&lt;Password&gt;</code></td>
</tr>
</tbody>
</table>

3. Change the password then Sign In.
4. Navigate to the below work center view.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESS CONFIGURATION</td>
<td>IMPLEMENTATION PROJECTS</td>
</tr>
</tbody>
</table>

5. Select First Implementation project.
6. Choose Open Activity List.
7. Choose Define Administrators for Project Team in Prepare.
8. Choose Lock Initial User.

10. Click OK and close the opened pages.
1.6.1.2.2.1.4 Scoping with Admin User

1. Choose work center view:

<table>
<thead>
<tr>
<th>Work Center</th>
<th>BUSINESS CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>IMPLEMENTATION PROJECTS</td>
</tr>
</tbody>
</table>

2. Select First Implementation project.
3. Choose Edit Project Scope button.
4. In the Country and Type of Business step, choose Edit Countries.

5. Add US to the Selected Countries and choose OK.
6. Choose Next.
7. In the Implementation Focus step, SAP Cloud for Customer is selected. Choose Next.

<table>
<thead>
<tr>
<th>Implementation Focus</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Cloud for Customer</td>
<td>Checked</td>
</tr>
</tbody>
</table>

8. In the Scoping step, maintain the following values and choose Next:
### Scoping Element L1 | Scoping Element L2 | Scoping Element L3
---|---|---
Sales | Account and Activity Management | Account Management  
| | | Activity Management  
| | | Communication for Account and Activity Management  
| | | Analysis for Account and Activity Management  
Sales | Product and Service Portfolio for Sales | <all>  
| | | Expand the scoping element and check the sub element “Sell Standard Products”.  
Sales | New Business | Opportunities  
| | | Sales Quotes  
| | | Sales Orders  
Business Performance Management | <all> | <all>  
Communication and Information Exchange | Business Process Management | <all>  
Communication and Information Exchange | Integration with External Applications and Solutions | Integration Scenarios with Cloud Solutions from SAP  
General Business Data | Business Partners | <all>  
General Business Data | Employees and Service Agents | <all>  
General Business Data | Products | Product Category hierarchy  
General Business Data | Products | Materials  
General Business Data | Products | Communication for Product Master Data  
General Business Data | Products | Analysis for Product Master Data  
General Business Data | Product and Service Pricing | <all>  
Built-in Services and Support | <all> | <all>  

#### i Note
Scoping depends on the business needs. Some of the elements are selected automatically due to business logic. This is needed for Communication and Information Exchange.

9. Choose Next.
10. In the Questions step, navigate using the following path.
   - Communication and Information Exchange ➔ Integration with External Applications and Solutions ➔
11. Select the Scoping element, on the right pane maintain the following values.
12. Mark *In Scope* for below elements.

<table>
<thead>
<tr>
<th>Scoping Element</th>
<th>Business Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration Scenarios with Cloud Solutions from SAP</td>
<td>Integration with SAP CPQ: Do you want to create and view sales quotes in SAP CPQ application from with your cloud solution?</td>
</tr>
</tbody>
</table>

13. In the *Questions* step, navigate using the following path.

- General Business Data ➔ Product and Service Pricing ➔ Sales Price specifications

14. Mark *In Scope* for below elements:

<table>
<thead>
<tr>
<th>Scoping Element</th>
<th>Business Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Business Data ➔ Product and Service Pricing</td>
<td>Do you want to maintain prices without price lists that will be used for offline pricing?</td>
</tr>
<tr>
<td>Sales Price specifications</td>
<td></td>
</tr>
</tbody>
</table>
15. Choose Next.
16. Choose Yes on the pop-up to confirm the values selected in the Questions step.
17. Review the Scoping Setting and choose Finish to complete the Scoping.

1.6.1.2.2 Fine Tuning

This section describes the need to perform fine-tuning to support specific business requirements.

Perform the following fine-tuning to adjust the system settings to support specific business requirements.

Define Business Roles [page 138]
Master Data Preparation Manually [page 140]
Configure SAP CPQ URL [page 141]
Define Communication System [page 142]
Configure Communication Arrangement [page 143]

1.6.1.2.2.2 Define Business Roles

1. Navigate to below work center.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>ADMINISTRATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>GENERAL SETTINGS</td>
</tr>
</tbody>
</table>

2. Select Business Roles under the section Users.
3. Choose New Business Role.

4. Maintain Business Role ID and Click on Maintain Name and Description button.
5. Choose the **WORK CENTER AND VIEW ASSIGNMENTS** tab.
6. Select below assignment.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERAL</td>
<td>Business Role ID</td>
<td>SALES_ROLE</td>
</tr>
<tr>
<td>GENERAL</td>
<td>Business Role Name</td>
<td>Sales Rep</td>
</tr>
<tr>
<td>GENERAL</td>
<td>Description</td>
<td>Role for Sales Representative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tab</th>
<th>Work Center/View ID</th>
<th>Work Center/View Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WORK CENTER AND VIEW ASSIGNMENTS</strong></td>
<td>CODACCOUNTWCV</td>
<td>Customers Accounts</td>
</tr>
<tr>
<td></td>
<td>COD_CONTACT_WCV</td>
<td>Customers Contacts</td>
</tr>
<tr>
<td></td>
<td>CODACCHIERWCV</td>
<td>Customers Accounts Hierarchy</td>
</tr>
<tr>
<td><strong>WORK CENTER AND VIEW ASSIGNMENTS</strong></td>
<td>COD_PRODUCTDATA COD_MATERIAL</td>
<td>Products</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Products Prices</td>
</tr>
<tr>
<td><strong>WORK CENTER AND VIEW ASSIGNMENTS</strong></td>
<td>COD_SALES_WCF</td>
<td>Sales</td>
</tr>
<tr>
<td></td>
<td>COD_OPPORTUNITY_WCVVIEW</td>
<td>Sales Opportunities</td>
</tr>
<tr>
<td></td>
<td>COD_QUOTE_WCVVIEW</td>
<td>Sales Quotes</td>
</tr>
<tr>
<td></td>
<td>COD_SALESORDER_WCVVIEW</td>
<td>Sales Orders</td>
</tr>
<tr>
<td><strong>WORK CENTER AND VIEW ASSIGNMENTS</strong></td>
<td>COD_ACTIVITIES</td>
<td>Activities &lt;All&gt;</td>
</tr>
</tbody>
</table>

7. Select **Action**  **Activate**
8. **Save** and **Close** the business roles page.
1.6.1.2.2.2 Master Data Preparation Manually

Add Activities into Project for Integration

1. Navigate to the following work center view.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>BUSINESS CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>IMPLEMENTATION PROJECTS</td>
</tr>
</tbody>
</table>

2. Choose Open Activity List.
4. Select All Activities in the Show dropdown field.
5. Add below activities to the project scope by selecting them and choose Add to Project.

<table>
<thead>
<tr>
<th>Activities</th>
<th>Activity Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Quotes</td>
<td>Sales</td>
</tr>
<tr>
<td>Code List Mapping for integration with External Applications and Solutions</td>
<td>Communication and Information Exchange</td>
</tr>
<tr>
<td>Communication Language for Data Replication</td>
<td>Communication and Information Exchange</td>
</tr>
<tr>
<td>Configure Price Strategy</td>
<td>General Business Data</td>
</tr>
</tbody>
</table>

6. Choose Close.

Communication Language for Data Replication

1. Navigate to the following work center view.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>BUSINESS CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Choose Open Activity List.
4. Choose Communication Language for Data Replication in the table.
5. Choose Add Row. And add your default language for text replication. Example: EN
6. Click Save and Close.

**Configure Sales Quote**

1. Navigate to the following work center view.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>BUSINESS CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>IMPLEMENTATION PROJECTS</td>
</tr>
</tbody>
</table>

2. Choose Open Activity List.
4. Choose Sales Quotes.
5. Choose Maintain Document Types.
6. Choose Add Row.
7. Enter ZCPQ in the Document Type Field and enter Description as Configure Price Quote.
8. Choose Direct Customer Acceptance, External Pricing and CPQ check box for this newly added document type.
9. Click Save and Close.
10. Choose Close

**1.6.1.2.2.2.3 Configure SAP CPQ URL**

1. Navigate to the following work center view.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>ADMINISTRATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>SALES AND CAMPAIGN SETTINGS</td>
</tr>
</tbody>
</table>

2. Choose Open Configure Price Quote Administrator under Configure Price Quote Settings.
1.6.1.2.2.2.4 Define Communication System

1. Navigate to the following work center view.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>ADMINISTRATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>GENERAL SETTINGS</td>
</tr>
</tbody>
</table>


3. Choose New to define Communication System, maintain as following.

   ![Diagram of Communication System setup]

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>&lt;CPQ_DEV&gt;</td>
<td></td>
</tr>
<tr>
<td>SAP Business Suite</td>
<td>Uncheck</td>
<td></td>
</tr>
<tr>
<td>Host Name</td>
<td>Maintain CPQ host URL</td>
<td>Host Name is the value CPQ URL without prefix HTTPS://</td>
</tr>
<tr>
<td></td>
<td>example: xxxx.webcomcpq.com</td>
<td></td>
</tr>
<tr>
<td>System Access Type</td>
<td>Internet</td>
<td></td>
</tr>
<tr>
<td>INTERNAL COMMENT</td>
<td>Internet</td>
<td>Optional</td>
</tr>
<tr>
<td>Preferred Application Protocol</td>
<td>5-Web Service</td>
<td></td>
</tr>
</tbody>
</table>

4. Maintain the person responsible for system maintenance in the TECHNICAL CONTACT.
5. Click Save and then choose Actions ➤ Set to Active.
6. Choose Close.

1.6.1.2.2.2.5 Configure Communication Arrangement

Communication scenarios mentioned in the table below are relevant for your project scope, create them one after the other:

<table>
<thead>
<tr>
<th>Communication Scenarios</th>
<th>Communication Services Inbound</th>
<th>Communication Services Outbound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Quote Replication from SAP Business Suite</td>
<td>• Replicate Sales Quote from SAP Business Suite ➤ Select</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

In case you do not see any of the configuration scenarios, re-check the project scoping to add the necessary scoping elements.

1. Navigate to ➤ Administrator ➤ General Settings ➤ Integration ➤ Communication Arrangements.
2. Choose **New** to create a new communication arrangement.
3. In **Communication Scenarios** table, select the **Communication Scenario Sales Quote Replication** from **SAP Business Suite**.
4. Choose **Next**.
5. Choose **System Instance ID** from the value help. You have created this communication system ID of the SAP S/4HANA system in the **Define Communication System** chapter.
6. Choose **Code List Mapping** as **No Mapping**.

7. Choose **Next**.
8. Inbound Settings
   For **INBOUND COMMUNICATION BASIC SETTINGS**, choose the **Application Protocol** as **Web Service** and **Authentication method** as **User ID and Password**.

9. Click **Edit Credentials**.
10. On the **Change Password** tab, Enter the password for above mentioned user. The same password is to be used in CPI.
11. Choose *Edit Advanced Settings*.

12. Under the *Inbound tab*, only select *Replicate Sales Quote from SAP Business Suite*.

13. Outbound Settings

For *OUTBOUND COMMUNICATION CREDENTIALS* under Technical Data tab, Disable Outbound communication.

### 1.6.1.2.2.3 Configure Single Sign-On in SAP Cloud for Customer

This section describes how to access the target SAP CPQ system without a need to enter the login credentials.

**Purpose**

With Single Sign-On enabled for both SAP Cloud for Customer and SAP CPQ system using a common Identity provider, business user can access the target SAP CPQ system without a need to enter the login credentials.

**Pre-requisite**

You should have a license of SAP Cloud Platform Identity Authentication Services.

**Procedure**

1. Login to your SAP Cloud Identity Provider account with your admin user.
2. From the Left Panel, Navigate to *Application and Resources ➤ Tenant Settings ➤*.
3. Choose *SAML 2.0 Configuration*.
4. On the new Screen, Click on *Download Metadata File*.
5. Login to your SAP Cloud for Customer system.
6. Navigate to the following work center view.

<table>
<thead>
<tr>
<th>Work Center</th>
<th>ADMINISTRATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>COMMON TASK</td>
</tr>
</tbody>
</table>


**i Note**
This Metadata file is needed when you setup the trust in your SAP Cloud Identity Provider.

9. In the General Pane, check the Manual Identity Provider.
10. Under the section Single Sign-On URL Handling, select the URL sent to Employee as Non-SO URL and SSO URL.

11. Choose Identity Provider tab.
12. Click on New Identity Provider tab. In the pop-up window, import the Metadata file of your SAP Cloud identity provider.

13. Enter an Alias name for your SAP Cloud identity provider.
14. Click Save.
15. Choose Activate Single Sign-On and choose OK on the pop-up dialog box that appears.
16. Login to your SAP Cloud Identity Provider account with your admin user.
17. Click on Applications tile to create your SAP Cloud for Customer application.
18. Click on Add and enter your SAP Cloud for Customer system name.
   For Example: C4C-CPQ
19. Click Save.
20. In the new Screen, Click on Home URL and enter your C4C SSO URL.
   For Ex: https://<yourTenant>.sso.vlab.sapbydesign.com
21. Click Save.
22. In the Trust, Choose SAML 2.0 Configuration and Import the C4C metadata file you have downloaded earlier.
23. Click Save.
24. Configure Name ID Attribute as Login Name.
25. Click Save.
26. Configure Default Name ID Format as Unspecified.
27. Click Save.
29. Create a CSV file for all the C4C user you want to enable SSO.
30. Choose Import.
31. Choose Send E-mails and Click Send.
32. Activation E-Mail to all the user will be triggered.

**i Note**
Once the user set their password for the IdP, they can login SAP Cloud for Customer using the SSO URL.

### 1.6.1.2.3 Configuration in SAP Cloud Platform Integration

This section describes the steps for configuring the integration of SAP CPQ with SAP Cloud for Customer using SAP Cloud Platform Integration (CPI).

**i Note**
SAP provides prepackaged, generic integration content called integration flows (iFlows) for the integration of SAP CPQ with SAP Cloud for Customer using SAP Cloud Platform Integration (CPI).

#### Prerequisites

To be able to import and deploy iFlows, you need the AuthGroup.IntegrationDeveloper role assigned in your tenant. Upload SAP CPQ SSL Certificates in SAP Cloud Platform Integration.

#### 1.6.1.2.3.1 How to get the SAP CPI client certificate

In the SAP CPI provisioning e-mail follow the link under Certificate Information. In the resulting screen choose X509 Certificate with option Binary CER. Download it to your local machine.
1.6.1.2.3.2 View prepackaged iFlows using SAP CPI Web UI

1. Access the web UI URL from the provisioning e-mail. It should be in the format: https://<hcitenant>.hana.ondemand.com/itspaces.
2. View all pre-packaged integration flows delivered by SAP in the Discover tab.
3. Click on the package SAP Cloud for Customer Integration with SAP CPQ.
4. You can see all the artifacts for this Integration Package.

1.6.1.2.3.3 Creating User Credentials in CPI System

1. Connect to the tenant management node of the SAP CPI system with the URL http://<tenant management node URL>/itspaces
2. Go to the Monitor tab, then Click on Security Material under Manage Security.
3. Click Add and select User Credentials.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>COM_USER_C4C</td>
</tr>
<tr>
<td>User</td>
<td>_CPQ_DEV</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password maintained in Communication Arrangement in SAP C4C</td>
</tr>
</tbody>
</table>
5. Click **Deploy**.
6. In a similar way, create another user credential in CPI for ADMIN user of SAP Cloud for Customer.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>COM_USER_ADMIN</td>
</tr>
<tr>
<td>User</td>
<td>PROJECTADMIN</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password maintained for admin user in SAP Cloud for Customer</td>
</tr>
</tbody>
</table>

**1.6.1.2.3.4 Configure and Deploy the iFlows Using Web UI**

**i Note**

If you have re-deployed key store (system.jks) in your SAP CPI tenant, you need to re-deploy all the iFlows to avoid the caching issues or re-start them to avoid caching related issues.

**Procedure**

1. Connect to the tenant management node of the SAP CPI system with the URL.
   
   `http://<tenant management node URL>/itspaces`

2. Under the **Discover** tab, Click on the package SAP Cloud for Customer **Integration with** SAP CPQ.
3. Choose **Copy** on the top-right corner of the package overview page.
4. If the Integration package were being created for the first time, then you would see the message ‘Integration Package Created’. If not, you will see the below dialog box asking to either create a new copy of the package or to overwrite the existing integration package content. Choose **Overwrite**.
5. Select the **Design** mode to configure the iFlows.

6. Select the Integration Package copied.

7. All the iFlows will be shown/ listed under the **ARTIFACTS** tab of the package.

8. Choose the iFlows mentioned in the table below (one after the other) by choosing **Actions** > **Configure** option.

9. Configure the **sender system** (in this example CPQ) and **receiver system** (in this example C4C) details as explained below:
   1. Choose **Sender** Tab: Choose **Authorization** as **User Role**.
   2. Choose **Receiver** Tab: Enter the **host** name as C4C tenant’s URL (without https ://) and **port** as 443.
   3. Choose the **Proxy Type** as **Internet**.
   4. Choose the **Authentication** as **Basic**.
   5. Maintain the **Credential Name** as per the below table.
   6. Choose **Save** and Choose **Deploy**.
7. Repeat the same procedure for all the iFlows that have the same sender system as CPQ and Receiver System as C4C.

<table>
<thead>
<tr>
<th>Name of the iFlow</th>
<th>Description of the iFlow</th>
<th>Receiver User Credential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replicate Quote to SAP CPQ</td>
<td>Request Quote information from SAP Cloud for Customer</td>
<td>COM_USER_ADMIN</td>
</tr>
<tr>
<td>Replicate Quote from SAP CPQ</td>
<td>Update SAP Cloud for Customer Quote from SAP Sales CPQ</td>
<td>COM_USER_C4C</td>
</tr>
</tbody>
</table>

### 1.6.1.2.3.5 Value Mapping for C4C-CPQ integration

1. Connect to the tenant management node of the CPI system with the url: `http://<tenant management node>/itspaces`
2. Go to Design view.
3. Select the Integration Package in your workspace.
4. All the iFlows will be shown/ listed under the Artifacts section of the page.
5. Select the iFlows with name Value Mapping for SAP Cloud for Customer integration with SAP CPQ and choose Actions button on the right and select Configure. All the Mapping Values will be shown/ listed.
6. Map all the Value mapping fields as per below sections.
7. Choose Save and then Deploy.

**Region Code**

2. Enter the respective Region Code in CPQ system in right side. Maintain the States Abrev2 available in CPQ system under General as States.

Country Code

2. Enter the respective Country Code in CPQ system in right side. Maintain the Countries Abrev3 available in CPQ system under General as Countries.

Distribution Channel

1. Mention the Distribution Channel Code in SAP Cloud for Customer system on left side. You can check Distribution Channel in C4C in Distribution Channel Activity under Business Configuration.
2. Enter the respective Distribution Channel in CPQ system in right side. Maintain the Distribution Channel maintained in CPQ system in Pricebooks under Pricing/Calculations.

Unit of Measurement

1. Mention the Quantity Unit in SAP Cloud for Customer system on left side. You can check Quantity Unit in C4C in Quantities and Units of Measurement Activity under Business Configuration.
2. Enter the respective Unit of Measurement in CPQ system in right side. Maintain the Unit of Measurement available in CPQ system under Product Catalog as Unit of Measurement.
Order Status

1. Mention the Lifecycle Sales Quote Status in SAP Cloud for Customer system on right side.
2. Enter the respective Status Code in CPQ system in left side. Maintain the Status Code available in CPQ system under Workflow/Approval as Statuses.

1.6.1.2.4 Data Loading

Data Loading can take place as per the below diagram.
1.6.1.3 Manual Build Steps

1.6.1.3.1 Ticket Component

<table>
<thead>
<tr>
<th>Implementation Step</th>
<th>Component</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV-RDS-CFC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.6.1.4 User Journey from SAP Cloud for Customer to SAP CPQ

This topic describes a typical user journey for this integration.

The user journey starts from a sales quote in SAP Cloud for Customer where users can define the involved parties (customers in SAP CPQ), the document type, header fields, and they can add products. After clicking Save and Open, the user is redirected from SAP Cloud for Customer to SAP CPQ where a new quote is created. Clicking Save creates a sales quote in SAP Cloud for Customer and the next time user edits it, they are redirected to SAP CPQ and a new quote is created. The quote in SAP CPQ uses SAP Cloud for Customer external ID as quote composite number so users have the experience of working only on one quote.

Additionally, users landing from SAP Cloud for Customer cannot access the Load Existing Project/Quotation page, so they will always land on the quote that they have opened in SAP Cloud for Customer. Users landing from SAP Cloud for Customer also have access to the User Page and Quote Layout pages under the Setup icon.

Even though most of the infrastructure matches the infrastructure of the old SAP Cloud for Customer integration, the landing page was developed completely new. The landing page displays after landing from SAP Cloud for Customer, while a quote is being created in SAP CPQ. The page displays the progress of the quote creation and messages in case of an error.

The SAP Cloud for Customer sales quote is automatically updated on the Save Quote action only if the quote in SAP CPQ is in active revision. Otherwise, if the quote is in inactive revision, the system skips the automatic syncing of data if the Save Quote action is triggered. If Update Sales Quote is triggered in SAP CPQ, an error message is displayed because the sales quote cannot be updated from an inactive revision.

**Note**

Configurable products synchronized from SAP Variant Configuration and Pricing [page 78] to SAP Cloud for Customer can be used in a scenario for the SAP CPQ integration.

When synchronized configurable products are added to the SAP Cloud for Customer Sales Quote, and the user lands from SAP Cloud for Customer to SAP CPQ, configurable products are added to the SAP CPQ quote with the Incomplete Configuration status in the Quotation tab. The prices of these items are all 0. However, when a configuration is edited, the edited item is updated with the prices from SAP Variant Configuration and Pricing.
Adding Items [page 156]

The integrated SAP Cloud for Customer - SAP CPQ environment supports simple products. Firstly, the products need to be created in SAP ERP and then replicated in SAP CPQ and SAP Cloud for Customer.

Customers [page 157]

SAP Cloud for Customer sends data for the bill-to and ship-to customers to SAP CPQ.

Field Mapping [page 158]

This topic explains how fields are mapped between a sales quote in SAP Cloud for Customer and a quote in SAP CPQ.

Updates and Status Change [page 161]

You can map statuses in SAP Cloud Platform Integration. Changes made on the quote in SAP CPQ are sent to SAP Cloud for Customer automatically or by using the dedicated action.

Placing Order to ERP [page 163]

When the Place Order to ERP action is executed, SAP ERP processes the order.

Payloads for Quote Creation and Quote Synchronization in SAP Cloud for Customer [page 163]

This section contains two XSD schemas, one with the payload used for quote creation in SAP CPQ, and the other with the payload used for synchronization with the SAP Cloud for Customer sales quote.

1.6.1.4.1 Adding Items

The integrated SAP Cloud for Customer - SAP CPQ environment supports simple products. Firstly, the products need to be created in SAP ERP and then replicated in SAP CPQ and SAP Cloud for Customer.

Unit of Measure

The Unit of Measure Code is used only to validate that the products have the same unit of measure. If they differ, a warning message (Product with material id MaterialId could not be added. Unit of measure does not match.) will be displayed on the SAP CPQ quote and the product will not be added. This is a standard field on SAP products, maps to the Base Unit of Measure in sales quotes.

Item Custom fields

If an item custom field is in the invalid format, an appropriate error message will be displayed to users and the SAP CPQ quote will not be created. If an item custom field cannot be found, the SAP CPQ quote will be created and a warning message will be displayed to the user: Quote item custom field FieldName could not be set. Quote item custom field could not be found. All other properties will be created and populated properly.

When SAP CPQ attempts to map values received from SAP Cloud for Customer to custom fields or quote item custom fields in the form of autocomplete attributes, the value received from SAP Cloud for Customer is matched with the value of the column defined as Value Code in the definition of the autocomplete attribute in the related quote table or custom table. The autocomplete attribute is only populated if one matching row is found. If multiple rows are found, or if no matching rows are found, the attribute is not populated and an error message is displayed.
ExternalCartItem

Maps to Item ID - the sales quote item ID is preserved in SAP CPQ and forwarded to SAP ERP.

ProductSystemId

Maps to Material ID in SAP Cloud for Customer. ProductSystemId is used when finding product to add in SAP CPQ. If an empty ProductSystemId is given, such products will be ignored. Products are added to SAP CPQ according to SAP CPQ rules.

If a user does not have permission to add items, the SAP CPQ quote will not be created and an error message will be displayed: User is not allowed to add items to quote.

Depending on the reasons why users cannot add products to quotes, warning messages will display accordingly:

- Product must exist and must be visible to the user: Product with material id MaterialId could not be added. Product is not found.
- Product must not be replaced or discontinued: Product with material id MaterialId could not be added. Product is replaced or discontinued.
- Product must be simple or with a complete configuration: Product with material id MaterialId could not be added. Product configuration is not complete.
- Simple product part number must exist in selected pricebook (if that rule is enforced in SAP CPQ Setup): Product with material id MaterialId could not be added. Product does not exist in pricebook.

Related Information

User Journey from SAP Cloud for Customer to SAP CPQ [page 155]
Customers [page 157]
Field Mapping [page 158]
Updates and Status Change [page 161]
Payloads for Quote Creation and Quote Synchronization in SAP Cloud for Customer [page 163]

1.6.1.4.2 Customers

SAP Cloud for Customer sends data for the bill-to and ship-to customers to SAP CPQ.

An Account in SAP Cloud for Customer matches the Bill-to customer role in SAP CPQ and the Ship-to role in SAP Cloud for Customer matches the Ship-to customer role in SAP CPQ.

Additionally, the system validates the country and the state of the customer. The system checks whether the country exists in SAP CPQ as a 2-digit or 3-digit ISO country code. If the country does not exist, the quote is not created and a message stating that the customer country code is invalid is displayed.
Additionally, if the 2-digit code for the customer state does not exist for the country in SAP CPQ, an error stating that the state code is invalid is displayed. Since the format of the state and country codes is not the same in SAP Cloud for Customer and SAP CPQ, administrators need to map the statuses in SAP Cloud Platform Integration.

Support for Prospects

It is also possible to create quotes for prospects in SAP Cloud for Customer. When the quote creation is initiated by a user on the SAP Cloud for Customer side, the payload that is sent to SAP CPQ doesn’t contain the CustomerCode parameter because the system doesn’t have the SAP ERP ID of the prospect. Instead, the payload contains the CrmAccountId parameter with the ID of the prospect.

Consequently, in the SAP CPQ quote, the Customer ID field (mapped to the CustomerCode parameter) remains empty and Account Id is populated with the value of the CrmAccountId parameter. The same applies for the bill-to and the ship-to customers.

Once a prospect is converted to customer in SAP Cloud for Customer, a customer is created in SAP ERP and its ID is sent to SAP Cloud for Customer. From there, the customer ID along with the prospect ID is passed along to SAP CPQ and the following changes take place:

- The customer which was first created from the prospect with account ID is now updated with the SAP ERP ID. The same applies for ship-to and bill-to customers.
- All quotes created with the account ID are now updated with the customer ID. There is an API for updating quotes with the custome ID: api/v1/customers/c4c/update/{crmAccountId}.

Related Information

User Journey from SAP Cloud for Customer to SAP CPQ [page 155]
Adding Items [page 156]
Field Mapping [page 158]
Updates and Status Change [page 161]
Payloads for Quote Creation and Quote Synchronization in SAP Cloud for Customer [page 163]

1.6.1.4.3 Field Mapping

This topic explains how fields are mapped between a sales quote in SAP Cloud for Customer and a quote in SAP CPQ.

Quote ID

The ID of a sales quote is mapped with the quote composite number in SAP CPQ.
Opportunity Id

Contains the ID of the opportunity in SAP Cloud for Customer from which the quote in SAP CPQ was created, if any. This field is not editable in SAP CPQ.

Currency, Market and Pricebook

A Distribution Chain Code consists of the Sales Organization Code and the Distribution Channel Code separated by an underscore. The Sales Organization corresponds to the SAP CPQ Market Code and the Distribution Channel corresponds to the same name field on the SAP CPQ pricebook.

The currency contains a currency 3-digit ISO code. The Sales Organization and Currency are used for market determination, and then Distribution Channel is used for pricebook determination.

Market and pricebook resolution is done by respecting SAP CPQ visibility rules for markets and pricebooks. If multiple pricebooks match the criteria, they are sorted by the pricebook level and the one with the highest level (Level 1) is selected.

If the Distribution Chain is empty, the SAP CPQ default market and pricebook will be used, and if the Distribution Chain is provided, but not valid, an error will appear and the CPQ quote will not be created.

The following error messages can appear in connection to the Distribution Chain:

- Distribution Chain provided, but currency missing: Currency code is not provided.
- Distribution Chain in invalid format: Distribution Chain is not in valid format.
- Market not found: Matching market not found for the provided distribution chain and currency.
- Pricebook not found: Matching pricebook not found for the provided distribution chain and currency.
- Pricebook found, but not available to the current user: Pricebook matching the provided distribution chain and currency is not available.

Consequently, to enable users to create quotes, administrators should make these adjustments in the SAP CPQ Setup:

1. Create a currency that exists in SAP Cloud for Customer.
2. Create a market with the previously created currency. The market’s code should match the code of the Sales Organization in SAP Cloud for Customer.
3. Create a pricebook in the existing market. Manually enter the Distribution Channel Code on the pricebook. Pricebook entries are replicated via the API from SAP ERP.

Effective Date

ISO formatted date string (YYYY-MM-DD) that is mapped with the Pricing Date in SAP Cloud for Customer.
**DocumentTypeCode**

The quote standard field `DocumentTypeCode` is a four letters value (for example, ZCPQ) and refers to the quote document type in SAP Cloud for Customer. This field is not displayed on the user interface, but it can be fetched via the CTX, as `<CTX ( Quote.DocumentTypeCode ) *>`, and via scripting, as `Quote.DocumentTypeCode`. This field is not used in SAP CPQ, it is only forwarded to SAP ERP when the Place Order to ERP action is called.

**Origin**

The field `Origin` contains the name of the external system if a quote is related with an external quote. For quotes that contain SAP Cloud for Customer sales quote reference, the value is `C4C-SalesQuote`. The field is not visible on the user interface, but it is exposed through scripting as `Quote.ExternalQuoteOrigin` (retrieves the value of the field).

**Cart Comment**

Maps to `External Note` in SAP Cloud for Customer. SAP CPQ quote comment has a limit of 3000 characters, and if the received value is longer, it will be truncated.

**Custom Fields**

1. **PO Number** - contains the ID of the external document corresponding to the sales quote, for example, the `Order ID` from the customer's system. This custom field is editable in SAP CPQ. Maps to `External Reference`.
2. **Requested Date** - contains the requested delivery date connected with the shipping. It is an ISO formatted date string (YYYY-MM-DD). This custom field is editable in SAP CPQ. Maps to `Requested Date` in SAP Cloud for Customer.
3. **Quote Expiration Date** - maps to `Valid To` in SAP Cloud for Customer. A custom field in SAP CPQ that already exists on all tenants. It is an ISO formatted date string (YYYY-MM-DD). This custom field is editable in SAP CPQ.

If a custom field is in an invalid format, an appropriate error message will be displayed to the user and the SAP CPQ quote will not be created.

If a custom field cannot be found, the SAP CPQ quote will be created and a warning message will be displayed to the user: `Quote custom field FieldName could not be set. Custom field could not be found`. All the fields following the custom field that is not created in SAP CPQ (and therefore cannot be mapped) are created and populated properly. When the value for a required SAP CPQ custom field is not sent from SAP Cloud for Customer, the quote is created successfully, and all other properties are populated properly. The following message is displayed to the user: `Values for the following required SAP CPQ custom fields are not sent from SAP Cloud for Customer: (the missing values are listed).`
When SAP CPQ attempts to map values received from SAP Cloud for Customer to custom fields or quote item custom fields in the form of autocomplete attributes, the value received from SAP Cloud for Customer is matched with the value of the column defined as Value Code in the definition of the autocomplete attribute in the related quote table or custom table. The autocomplete attribute is only populated if one matching row is found. If multiple rows are found, or if no matching rows are found, the attribute is not populated and an error message is displayed.

**Additional Discount**

Additional discount in SAP CPQ is different from the header discount in SAP Cloud for Customer and therefore it is removed from SAP CPQ. The difference is that the additional discount in SAP CPQ is not propagated to items like the discount in SAP Cloud for Customer.

However, the discount feature is still enabled in SAP CPQ. Users can add a discount for product types which is then propagated to items. This value is reflected in the Overall Discount Percent field in the Subtotal section on quotes.

**Primary Quote Action**

If there are multiple quotes assigned to an opportunity, one of them can be marked as primary using the Make Primary action. The Primary Quote field is populated automatically with Yes and it’s read-only. All other quotes assigned to the same opportunity are automatically marked as Not primary.

After triggering the action, the payload for synchronization of the SAP Cloud for Customer quote is sent, including the ID of the opportunity, the total value of the quote and the indicator that the quote is made primary.

**Related Information**

- User Journey from SAP Cloud for Customer to SAP CPQ [page 155]
- Adding Items [page 156]
- Customers [page 157]
- Updates and Status Change [page 161]
- Payloads for Quote Creation and Quote Synchronization in SAP Cloud for Customer [page 163]

**1.6.1.4.4 Updates and Status Change**

You can map statuses in SAP Cloud Platform Integration. Changes made on the quote in SAP CPQ are sent to SAP Cloud for Customer automatically or by using the dedicated action.

Users cannot go to SAP Cloud for Customer from SAP CPQ (the workflow is one-directional from SAP Cloud for Customer to SAP CPQ only). All changes made on the quote in SAP CPQ are sent to SAP Cloud for Customer...
either automatically after users save the quote or when triggering the Update Sales Quote action. Additionally, if the Automatically Update Sales Quote parameter is enabled, whenever the quote status is changed in SAP CPQ, the sales quote in SAP Cloud for Customer is updated accordingly. Also, whenever any action is triggered on the quote, including Save, the system checks whether the status should be changed.

The mapping of statuses is done in SAP Cloud Platform Integration and it depends on the user’s business requirements. This table shows a typical mapping of statuses in SAP CPQ with the statuses in SAP Cloud for Customer, although you can map the statuses as needed:

<table>
<thead>
<tr>
<th>SAP CPQ Status</th>
<th>SAP Cloud for Customer Lifecycle Status</th>
<th>SAP Cloud for Customer Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparing</td>
<td>Open</td>
<td>Not relevant</td>
</tr>
<tr>
<td>Awaiting Internal Approval</td>
<td>Open</td>
<td>Not relevant</td>
</tr>
<tr>
<td>Approved</td>
<td>Open</td>
<td>Not relevant</td>
</tr>
<tr>
<td>Rejected</td>
<td>Open</td>
<td>Not relevant</td>
</tr>
<tr>
<td>Open</td>
<td>Open</td>
<td>Pending</td>
</tr>
<tr>
<td>Customer Accepted</td>
<td>Open</td>
<td>Pending</td>
</tr>
<tr>
<td>Customer Rejected</td>
<td>Open</td>
<td>Pending</td>
</tr>
<tr>
<td>Order Placed</td>
<td>Completed</td>
<td>Pending</td>
</tr>
<tr>
<td>Order FAILED</td>
<td>Open</td>
<td>Pending</td>
</tr>
<tr>
<td>Expired</td>
<td>Completed</td>
<td>Lost* (need to supply reason for rejection)</td>
</tr>
<tr>
<td>Lost</td>
<td>Completed</td>
<td>Lost* (need to supply reason for rejection)</td>
</tr>
</tbody>
</table>

If the status of the quote changes to a status in which the edit and view actions are not available (for example, when users send the quote for approval and the status changes), users are redirected to a blank page displaying a message that they don’t have the permission to view the quote.

Additionally, if the quote status is changed to a status in which users only have the permission to view the quote, but not to edit it, the quote will be opened in the view-only mode.

**Note**

When landing from SAP Cloud for Customer, the action Retract Approval Process can only be accessed via the Request Approval option in the quote. You will need to leave a comment to perform the action.

**Related Information**

- User Journey from SAP Cloud for Customer to SAP CPQ [page 155]
- Adding Items [page 156]
- Customers [page 157]
- Field Mapping [page 158]
- Payloads for Quote Creation and Quote Synchronization in SAP Cloud for Customer [page 163]
1.6.1.4.5  Placing Order to ERP

When the Place Order to ERP action is executed, SAP ERP processes the order.

Once the order is processed, SAP ERP calls the PerformCartAction API on the SAP CPQ side to execute the Order Status Update action, which changes the status according to the Workflow and sets an Order Id for the quote. In the recommended setting, the Order Status Update action changes the status from Order Confirmation Pending to Order Placed.

After the order is placed, changes made in SAP ERP are not visible to the user until the quote is reloaded. To reload the quote in SAP Cloud for Customer, refresh the browser (by pressing F5). To reload the quote in SAP CPQ, go to the quote list and reopen the quote.

After placing the order, the user shouldn’t make changes on the quote, and only the action Change Status should be available. In case there was an error on the SAP ERP side, use the Change Status action to change the status from Order Confirmation Pending to a status which allows the user to make changes on the quote. That way, you can fix the error and try to place the order again.

1.6.1.4.6  Payloads for Quote Creation and Quote Synchronization in SAP Cloud for Customer

This section contains two XSD schemas, one with the payload used for quote creation in SAP CPQ, and the other with the payload used for synchronization with the SAP Cloud for Customer sales quote.

GetSalesQuoteDetails Payload

```xml
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SalesQuoteDetails">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="QuoteId" minOccurs="1" maxOccurs="1"/>
        <xs:element type="xs:string" name="QuoteName" minOccurs="0" maxOccurs="1" nillable="true"/>
        <xs:element type="xs:string" name="DistributionChain" minOccurs="1" maxOccurs="1"/>
        <xs:element type="xs:string" name="Currency" minOccurs="0" maxOccurs="1" nillable="true"/>
        <xs:element type="xs:string" name="PricingDate" minOccurs="0" maxOccurs="1" nillable="true"/>
        <xs:element type="xs:string" name="EffectiveDate" minOccurs="0" maxOccurs="1" nillable="true"/>
        <xs:element type="xs:string" name="RequestedDate" minOccurs="0" maxOccurs="1" nillable="true"/>
        <xs:element type="xs:string" name="DocumentTypeCode" minOccurs="0" maxOccurs="1" nillable="true"/>
        <xs:element type="xs:string" name="DocumentTypeValue" minOccurs="0" maxOccurs="1" nillable="true"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```
<xs:element type="xs:string" name="Comment" minOccurs="0" maxOccurs="1" nillable="true"/>
<xs:element type="xs:string" name="IsPrimary" minOccurs="0" maxOccurs="1" nillable="true"/>
<xs:element type="xs:string" name="OpportunityId" minOccurs="0" maxOccurs="1" nillable="true"/>
<xs:element type="xs:string" name="OpportunityName" minOccurs="0" maxOccurs="1" nillable="true"/>
<xs:element type="xs:string" name="OriginQuoteId" minOccurs="0" maxOccurs="1" nillable="true"/>
<xs:element name="CustomFields" minOccurs="0" maxOccurs="1" nillable="true">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CustomField" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="Name" minOccurs="1" maxOccurs="1"/>
            <xs:element type="xs:string" name="Value" minOccurs="1" maxOccurs="1" nillable="true"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Items" minOccurs="0" maxOccurs="1" nillable="true">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Item" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="ItemId" minOccurs="1" maxOccurs="1"/>
            <xs:element type="xs:string" name="Quantity" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="ProductSystemId" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="UnitOfMeasureCode" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Comment" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element name="CustomFields" minOccurs="0" maxOccurs="1" nillable="true">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="CustomField" maxOccurs="unbounded" minOccurs="0">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element type="xs:string" name="Name" minOccurs="1" maxOccurs="1"/>
                        <xs:element type="xs:string" name="Value" minOccurs="1" maxOccurs="1" nillable="true"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="BillTo" minOccurs="0" maxOccurs="1" nillable="true">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string" name="FirstName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="LastName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Title" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CompanyName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="AddressLine1" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="AddressLine2" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="City" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Province" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="State" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Country" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Territory" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="ZipCode" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="PrimaryIndustry" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="TerritoryName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Phone" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Fax" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Email" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CustomerCode" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CrmAccountId" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CrmContactId" minOccurs="0" maxOccurs="1" nillable="true"/>
    </xs:sequence>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="CustomFields" minOccurs="0" maxOccurs="1" nillable="true">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CustomField" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="Name" minOccurs="1" maxOccurs="1" nillable="true"/>
                    <xs:element type="xs:string" name="Value" minOccurs="1" maxOccurs="1" nillable="true"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:complexType>
<xs:complexType>
<xs:sequence>
  <xs:element type="xs:string" name="FirstName" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="LastName" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="Title" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="CompanyName" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="AddressLine1" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="AddressLine2" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="City" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="Province" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="State" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="Country" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="Territory" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="ZipCode" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="PrimaryIndustry" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="TerritoryName" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="Phone" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="Fax" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="Email" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="CustomerCode" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="CrmAccountId" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element type="xs:string" name="CrmContactId" minOccurs="0" maxOccurs="1" nillable="true"/>
  <xs:element name="CustomFields" minOccurs="0" maxOccurs="1" nillable="true">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="CustomField" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Name" minOccurs="1" maxOccurs="1" nillable="true"/>
              <xs:element type="xs:string" name="Value" minOccurs="1" maxOccurs="1" nillable="true"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:element name="SalesArea" minOccurs="0" maxOccurs="1" nillable="true">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string" name="SalesOrganization" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="DistributionChannel" minOccurs="1" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Division" minOccurs="0" maxOccurs="1" nillable="true"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="OrderStatus" minOccurs="0" maxOccurs="1" nillable="true">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string" name="Id" minOccurs="1" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Name" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="NameTranslated" minOccurs="0" maxOccurs="1" nillable="true"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="SelectedMarket" minOccurs="1" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string" name="CurrencyCode" minOccurs="1" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CurrencyDescription" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CurrencyRate" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CurrencySign" minOccurs="1" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="ForwardCurrencyRate" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="MarketCode" minOccurs="1" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="MarketFactor" minOccurs="1" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="MarketName" minOccurs="1" maxOccurs="1" nillable="true"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="CustomFields">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CustomField" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="Content" minOccurs="1" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Id" minOccurs="1" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Name" minOccurs="1" maxOccurs="1" nillable="true"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ShipToCustomer" minOccurs="0" maxOccurs="1" nillable="true">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string" name="QuoteId" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Active" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="FirstName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="LastName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="OwnerName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Title" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CompanyName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Address1" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Address2" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="City" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Province" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="StateAbbreviation" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CountryAbbreviation" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="ZipCode" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="PrimaryIndustry" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="TerritoryName" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="BusinessPhone" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="BusinessFax" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="Email" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CustomerCode" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CustomerType" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CrmAccountId" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="CrmContactId" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="UserId" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element type="xs:string" name="ExternalId" minOccurs="0" maxOccurs="1" nillable="true"/>
      <xs:element name="CustomFields" minOccurs="0" maxOccurs="1" nillable="true"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EndUserCustomer" minOccurs="0" maxOccurs="1" nillable="true">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:string" name="QuoteId" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Active" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="FirstName" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="LastName" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="OwnerName" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Title" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="CompanyName" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Address1" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Address2" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="City" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Province" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="StateAbbreviation" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="CountryAbbreviation" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="ZipCode" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="PrimaryIndustry" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="TerritoryName" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="BusinessPhone" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="BusinessFax" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="Email" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="CustomerCode" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="CustomerType" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element type="xs:string" name="CrmAccountId" minOccurs="0" maxOccurs="1" nillable="true"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
Related Information

Specific SAP CPQ Prerequisites [page 129]
Field Mapping [page 158]

1.6.2 Legacy SAP Cloud for Customer - SAP CPQ Integration

This section describes the legacy SAP Cloud for Customer - SAP CPQ integration that was available to users who set up the integration before the 1905 SAP CPQ release. This integration became obsolete in 1905 for new users, although old users can use it uninterruptedly.

The improved SAP Cloud for Customer - SAP CPQ integration is described in SAP Cloud for Customer - SAP CPQ Integration [page 119].

(Legacy) SAP Cloud for Customer Integration Setup on the SAP CPQ Side [page 178]
To integrate SAP CPQ with SAP Cloud for Customer, you first need to enable the SAP provider and define common settings.

(Legacy) Scripting Support for the SAP Cloud for Customer Integration [page 180]
For SAP CPQ - SAP Cloud for Customer integration purposes, specific IronPython methods that can be called over SAP CPQ’s standard IronPython scripting functionality are exposed.
1.6.2.1 (Legacy) SAP Cloud for Customer Integration Setup on the SAP CPQ Side

To integrate SAP CPQ with SAP Cloud for Customer, you first need to enable the SAP provider and define common settings.

All communication from SAP CPQ to SAP Cloud for Customer goes over the SAP Cloud Platform Integration, a BUS service that orchestrates messages in different formats and has many adapters and connectors for different systems and protocols. REST API that processes SAP CPQ requests was developed on SAP Cloud Platform Integration.

Enabling SAP Provider and Defining Common Settings

1. Go to Setup Providers Providers SAP. Available SAP providers display.
3. On the Common Settings tab, turn on the Enable integration toggle switch.
4. In Hana Cloud Platform Integration - Authentication details, specify the following:
   ○ Authentication mode: the method for tracking and authenticating users. You can select either Hybrid, BasicAuth or SamiBearer.
   ○ Integration username: the username of the integration user.
   ○ Integration password: the password of the integration user.
5. In General settings, define the following:
   ○ Send customers along with quote payload: when the toggle switch is enabled, SAP CPQ sends customer details such as bill-to, ship-to and end user details, together with the quote data to the SAP system. The quote data is sent in XML format.
   ○ Send prices in the selected market: when the toggle switch is enabled, SAP CPQ sends prices calculated for the current quote’s market. If the toggle switch is disabled, prices are calculated for the default market.
   ○ Dates in quote standard fields (e.g. Date Created, Date Modified…) will be converted into this time zone: You can select either UserZone or UtcZone.
6. In SAP Cloud Platform Integration Service API Details, enter the following:
   ○ REST API base URL address: the base REST API URL address of the SAP Cloud Platform Integration – SAP CPQ specific package. The value is delivered by the administrator and should not be changed on the fly. In addition, the value identifies a specific SAP Cloud Platform IntegrationSAP Cloud Platform Integration environment, that is, the tenant.
   ○ Endpoint that gets SAP object data details: a relative endpoint address of the SAP Cloud Platform Integration artifact that gets an object from the SAP system over SAP Cloud Platform Integration. The call acts as a proxy to SAP OData services and can query any publicly visible object from SAP Cloud for Customer. To view examples of the method, please refer to the GetObjectData Endpoint.
   ○ Endpoint that processes posted data in xml format: a relative endpoint of the SAP Cloud Platform Integration artifact that processes the posted data in the XML format.
   ○ Endpoint that sends CPQ quote to SAP systems: a relative endpoint of the SAP Cloud Platform Integration artifact that sends the SAP CPQ quote to SAP systems.
7. Click Save.

**Defining SAP Cloud for Customer Settings**

On the C4C CRM Settings tab, provide SAP Cloud Platform Integration API details.

The following list explains the SAP Cloud Platform Integration REST API details.

- **Endpoint that gets opportunity details in case when landing and creating a new quote**: a relative endpoint address of the SAP Cloud Platform Integration artifact that retrieves Opportunity header details and Customer data (Bill To, Ship To and end User) in XML format. The artifact is called upon landing from the SAP Cloud for Customer Opportunity on CPQ for the first time. In addition, the action is also triggered when a new Quote is created. To learn how the artifact is created and how it functions, refer to the Opportunity Query Interface PDF file.

- **Endpoint that gets opportunity details in case when landing to existing quote**: a relative endpoint address of the SAP Cloud Platform Integration artifact that retrieves Opportunity header details and Customer data (bill to, ship to and end user) in XML format. The artifact is called upon landing from the SAP Cloud for Customer Opportunity on the already-existing Quote in CPQ.

- **Endpoint that updates SAP opportunity from CPQ quote data**: a relative endpoint address of the SAP Cloud Platform Integration artifact that processes the Quote that CPQ sends in XML format. In addition, the endpoint updates a remote SAP Cloud for Customer Opportunity that is linked to the CPQ Quote. Note that Opportunity is linked to the CPQ Quote after landing on CPQ from the SAP Cloud for Customer Opportunity. CPQ manages the relationship between the Quote and Opportunity. To learn how the artifact is created and how it functions, refer to the Opportunity Update Interface PDF file.

- **Endpoint that creates or updates SAP Sales Quote**: a relative endpoint address of the SAP Cloud Platform Integration artifact that processes the Quote that CPQ sends in XML format. Moreover, the endpoint creates and updates a remote SAP Cloud for Customer Sales Quote. Note that the Sales Quote is linked to the CPQ Quote after calling upon this action for the first time. CPQ manages the relationship between the Sales Quote and CPQ Quote. To learn how the artifact is created and how it functions, refer to the Sales Quote Creation Interface PDF file.

- **Endpoint that update SAP opportunity and creates or updates SAP Sales Quote**: a relative endpoint address of the SAP Cloud Platform Integration artifact that consolidates the previous two artifacts - Endpoint that updates SAP opportunity from CPQ quote data and Endpoint that creates or updates SAP Sales Quote.

- **Endpoint that attach document to the both SAP opportunity and sales quote**: a relative endpoint address of the SAP Cloud Platform Integration artifact that consolidates the following two artifacts - Endpoint that attach document to the SAP opportunity and Endpoint that attach document to the SAP sales quote. To learn how the artifact is created and how it functions, refer to the Attachment Creation Interface PDF file.

- **Endpoint that attach document to the SAP opportunity**: a relative endpoint address of the SAP Cloud Platform Integration artifact that attaches a sent CPQ Quote proposal document to a remote SAP Cloud for Customer Opportunity.

- **Endpoint that attach document to the SAP sales quote**: a relative endpoint address of the SAP Cloud Platform Integration artifact that attaches a sent CPQ Quote proposal document to a remote SAP Cloud for Customer Sales Quote. Please note that for the artifact’s execution to be successful, the Sales Quote should be created on a remote SAP Cloud for Customer system.

- **Endpoint that gets SAP object data details**: a relative endpoint address of the SAP Cloud Platform Integration artifact that gets an object from the SAP system over SAP Cloud Platform Integration (HCI). The call acts as a proxy to SAP OData services and can query any publicly visible object from SAP Cloud for Customer. To view examples of the method, refer to the GetObjectData Endpoint PDF file.
When all the SAP Cloud Platform Integration REST API details are set up, click **Save**.

**Related Information**

Legacy SAP Cloud for Customer - SAP CPQ Integration [page 177]

### 1.6.2.2  **(Legacy) Scripting Support for the SAP Cloud for Customer Integration**

For SAP CPQ - SAP Cloud for Customer integration purposes, specific IronPython methods that can be called over SAP CPQ’s standard IronPython scripting functionality are exposed.

The following is a list of available SAP scripting methods:

- **SAP.GetOpportunityDetailsForNewQuote(Quote.OpportunityId)**: gets the Opportunity header and Customer details from the SAP system for the newly created Quote in SAP CPQ that is associated with the incoming Opportunity ID.
- **SAP.GetOpportunityDetailsForExistingQuote(Quote.OpportunityId)**: gets the Opportunity header and Customer details from the SAP system for the existing SAP CPQ Quote that is associated with the incoming Opportunity ID.
- **SAP.UpdateOpportunity(Quote.OpportunityId)**: updates the Opportunity header and inserts new items from the SAP CPQ Quote payload. This scripting method should be used when only Opportunity needs to be updated, not the Sales Quote.
- **SAP.CreateUpdateSalesQuote(Quote.OpportunityId)**: creates/updates the Sales Quote in the SAP system. This is an UPSERT operation. If the Sales Quote does not exist for the provided Opportunity ID, this scripting method creates a new Sales Quote and inserts header and Sales Quote lines by using data from the input Quote. If the Sales Quote exists, the scripting method removes all the items and inserts new items from the CPQ Quote payload.
- **SAP.UpdateOpportunityAndSalesQuote(Quote.OpportunityId)**: updates both the Opportunity and Sales Quote in the SAP system. In addition, this scripting method aggregates **UpdateOpportunity** and **UpdateSalesQuote** artifacts. If possible, the artifact should call for a parallel creation/update of the Opportunity and the Sales Quote on the SAP Cloud Platform Integration side. The call may be used for sending documents and/or attachments from SAP CPQ to SAP or for setting the SAP CPQ Quote status to **Won**.
- **SAP.AttachDocumentToOpportunity(Quote.OpportunityId)**: attaches an SAP CPQ document (PDF, DOCX or XLSX) to the SAP Opportunity object.
- **SAP.GetQuoteInXml()**: gets a complete Quote serialized in XML format.

(1.6.2.2) **Creating Custom Actions through Scripting** [page 181]

The following two examples illustrate how to create custom actions through scripting in SAP CPQ and link them to other actions as a post and pre-action.

(1.6.2.2) **Sending Custom Data from SAP CPQ to SAP Cloud for Customer** [page 182]

When sending Quote data to a remote SAP Cloud for Customer system, SAP CPQ sends the complete Quote in XML format. The Administrator can easily add custom XML structure to the Quote XML if needed.
1.6.2.2.1 (Legacy) Creating Custom Actions through Scripting

The following two examples illustrate how to create custom actions through scripting in SAP CPQ and link them to other actions as a post and pre-action.

The User can change the names of custom actions. For demonstration purposes, Update Opportunity_Script and Create/Update Sales Quote_Script are used as names.

**i Note**
Creating custom actions through scripting is possible only in the Legacy SAP CPQ - SAP Cloud for Customer Integration.

**Update Opportunity as a Post-Action upon the Save Quote Action**

The example demonstrates how to create a custom action for updating an Opportunity and link it to the Save Quote action as a post-action. As a result, once a User clicks Save Quote, and the Save Quote action is executed, the Update Opportunity action is triggered.

In SAP Cloud for Customer, open an Opportunity and click the SAP CPQ tab.

To access custom actions from the SAP CPQ Quote, go to Setup > Develop > Custom Actions.

The Custom Actions page displays all previously created custom actions.

To create a new custom action (UpdateOpportunity_Script), do the following:

1. Click Add New.
2. Enter a name (UpdateOpportunity_Script) in the Name field. Please note that this field is required. In addition, System ID is created automatically based on the name you entered in the Name field.
3. (Optional) Fill in the fields you need to meet your business needs.
5. Click Save.

To link the newly created custom action (UpdateOpportunity_Script) to the Save Quote action as a post-action, take the following steps:

- Go to Workflow/Approval > Workflow.
  - In the My Quotes tab, click the icon in the intersection between the Open status and Open status.
  - Select the Save Quote action.
  - Click the Edit button in the PostActions column.
  - Set the previously created action (UpdateOpportunity_Script) as a post action.
    - Click Update.
  - Click Save on the Status Administration page.
  - Open the SAP CPQ Quote.

By clicking Save Quote, the Update Opportunity custom action is executed as a post-action.
Create/Update Sales Quote as a Pre-Action upon the Attach Document To Sales Quote action

The example demonstrates how to create a custom action for creating/updating a Sales Quote and link it to the Attach Document To Sales Quote action as a pre-action. After following the steps outlined in the example, the User is able to attach a document to the SAP Sales Quote even if the Sales Quote is not yet created. Clicking Attach Document To Sales Quote creates a SAP Sales Quote and also adds the generated document to the SAP Sales Quote.

To create a new custom action (CreateUpdateSalesQuote_Script), do the following:

1. Repeat steps 1, 3 and 5 from the previous example.
2. Enter a name (CreateUpdateSalesQuote_Script) in the Name field.
3. Write SAP.CreateUpdateSalesQuote(Quote.OpportunityId) in the Script field.

To link the newly created custom action (CreateUpdateSalesQuote_Script) to the Attach Document To Sales Quote action as a pre-action, take the following steps:

- Go to Workflow/Approval.
  - In the My Quotes tab, click the icon in the intersection between the Open status and Open status.
  - Select the Attach Document To Sales Quote action.
  - Click the Edit button in the PreActions column.
- Set the previously created action (CreateUpdateSalesQuote_Script) as a pre-action.
  - Click Update.
- Click Save on the Status Administration page.
- Go to the SAP CPQ Quote.
- Click Generate Documents.
- Choose a template and generate it, as explained in Create Document Generation Template.

By clicking Attach Document To Sales Quote, the Create/Update Sales Quote custom action is executed as a pre-action.

Related Information

(Legacy) Scripting Support for the SAP Cloud for Customer Integration [page 180]
Legacy SAP Cloud for Customer - SAP CPQ Integration [page 177]

1.6.2.2.2 (Legacy) Sending Custom Data from SAP CPQ to SAP Cloud for Customer

When sending Quote data to a remote SAP Cloud for Customer system, SAP CPQ sends the complete Quote in XML format. The Administrator can easily add custom XML structure to the Quote XML if needed.

With scripting, it is possible to create nodes, attributes and fill them with values. To send data, the User can use all SAP Create/Update actions with an additional method parameter.
- **Update Opportunity**: `SAP.UpdateOpportunity(Quote.OpportunityId, XmlNode)
- **Create/Update Sales Quote**: `SAP.CreateUpdateSalesQuote(Quote.OpportunityId, XmlNode)
- **Update Opportunity and Sales Quote**: `SAP.CreateUpdateSalesQuote(Quote.OpportunityId, XmlNode)

**Example**

The following example demonstrates how to include custom data in XML with the `UpdateOpportunityAndSalesQuote` action.

- Make a new custom action with a script that creates custom XML with XmlHelper.
- Write the following script:

```python
node1 = XmlHelper.CreateXmlNode("Street", "1234 Jefferson Street")
node2 = XmlHelper.CreateXmlNode("City", "San Diego")
node3 = XmlHelper.CreateXmlNode("State", "CA")
node4 = XmlHelper.CreateXmlNode("Postal", "90362")
nodeAddress = XmlHelper.CreateXmlNode("Address", node1, node2, node3, node4)
XmlHelper.SetAttribute(nodeAddress, "Name", "John")
SAP.UpdateOpportunityAndSalesQuote(Quote.OpportunityId, nodeAddress)
```

- Access the **Developer Console** from the **Setup** menu. In order for all methods and operations performed on the data to display, select **Full Mode**.
- Create a new Quote.
- Click the previously created custom action.
- Filter the list by entering **SAP** in the search field in the **Labels** column.
- Find all traces that contain **Sending data to SAP**.
- Click to view **Sending data to SAP** trace details. The Trace Details page appears.
  Custom XML created with XmlHelper is at the end of the Quote’s XML.

```xml
</PriceComponent>
<TotalAmount>0.00000</TotalAmount>
<QuoteTables />

<Custom>
  <Address Name="John">
    <Street>1234 Jefferson Street</Street>
    <City>San Diego</City>
    <State>CA</State>
    <Postal>90362</Postal>
  </Address>
</Custom>
</Quote>
```
iNote

When sending data to SAP Cloud for Customer, the default format for Date and DateTime fields is YYYY-MM-DD-HH.MM.SS.

<Quote>
  <CompositeNumber>01550016</CompositeNumber>
  <DateCreated>2017-12-28</DateCreated>
  <DateModified>2018-01-26</DateModified>
  <MarketFactor>1.000000</MarketFactor>
  <QuoteId>16</QuoteId>
  <OpportunityId>46554</OpportunityId>
  <AccountId>1002033</AccountId>
  <ExternalId />
  <Comment />
  <RevisionNumber>0</RevisionNumber>
  <UserId>155</UserId>
  <ExpirationDate>2018-01-27</ExpirationDate>
  <OrderStatus>
</Quote>

Sending Quote Tables from SAP CPQ to SAP Cloud for Customer

When sending quote payload to SAP Cloud for Customer, SAP CPQ sends quote tables.

XML generated for a quote table is at the end of the quote’s XML.

<iNote>
Even if a quote table is not visible to the logged-in user, SAP CPQ sends the quote table to SAP Cloud for Customer.
</iNote>
1.7  SAP Commerce Cloud Integration

The SAP CPQ – SAP Commerce Cloud integration increases the front office sales efficiency and enables B2B customers to directly interact with sales representatives via collaboration functionalities that both systems have.

*Note*

The integration described in this section is used with the Quote 1.0 engine. The corresponding integration used with the Quote 2.0 engine is described in SAP Commerce Cloud Integration for Quote 2.0 [page 191].

Prerequisites for SAP Commerce Cloud Integration [page 186]

These are the prerequisites which need to be completed before SAP CPQ can be integrated with SAP Commerce Cloud.

Integrate SAP CPQ with SAP Commerce Cloud [page 189]

Complete this procedure to enable the SAP Commerce Cloud integration on the SAP CPQ side.

SAP Commerce Cloud Integration Scenarios [page 189]

Depending on where the user journey starts, there are two use case scenarios: users can begin their journey from SAP Commerce Cloud, or from SAP CPQ.
1.7.1 Prerequisites for SAP Commerce Cloud Integration

These are the prerequisites which need to be completed before SAP CPQ can be integrated with SAP Commerce Cloud.

- In case SAP Commerce Cloud is using existing configurable products set up for the Variant Configuration Integration [page 78], SAP Variant Configuration and Pricing need to be integrated. However, this is not required for using simple products from SAP ERP.

- The SAP provider needs to be enabled and common settings [page 178] defined in Setup > Providers > SAP Cloud Platform > Common Settings.

- An email list needs to be created in Setup > Workflow > Email Lists. Since the email notification about the request for quotation (RFQ) is sent to sales representatives, we suggest you use Sales Representatives as the name of the email list. Additionally, when creating an email list, make sure you select the adequate user types (for example, Sales Representative).

- A notification that sales representatives receive about the request for quotation needs to be created in Setup > Workflow > Notifications. Example: For the notification name, enter RFQ Notification. For the message subject, you may use Request for Quotation. In Email List, select the previously created email list Sales Representatives. For the message body, you may use the following: Dear sales representative, there is a new quote with the number <SYSTEMQUOTATIONNUMBER > waiting for your review and proposal. Please follow the link to review the quote: < QuoteLink >
The following actions need to be added to the appropriate workflow matrix: Release Quote To SAP Commerce and Release Quote And Proposal To SAP Commerce. The former will be visible on quote, whereas the latter will be available upon document generation in the form of the Submit to customer button. Additionally, you need to set permissions for the said actions.

- The External Quote Status custom field needs to be created in Setup > Quotes > Custom Fields.
The integration user needs to be created in Setup > Users as the user will be the quote owner. The administrator role should be assigned to the user.

The New Revision action needs to be added to the workflow matrix. Additionally, the previously created integration user needs to have permissions for the above-mentioned action, as the action will be used to create a new revision on behalf of the integration user.

To successfully integrate any system with SAP CPQ, the application parameter Allowed origins for the CORS filter needs to be configured in the way explained in Application Parameters.

**Related Information**

System Requirements for SAP Integrations [page 5]
SAP CPQ Integration Module
1.7.2 Integrate SAP CPQ with SAP Commerce Cloud

Complete this procedure to enable the SAP Commerce Cloud integration on the SAP CPQ side.

Procedure

1. Go to Setup Providers SAP. Available SAP providers display.
3. On the Commerce Settings tab, enable the Connect SAP Commerce toggle switch.
4. From the Commerce Notification dropdown list, select the previously created RFQ notification as explained in Prerequisites for SAP Commerce Cloud Integration [page 186].
5. In Release Quote Endpoint, enter the endpoint that sends the quote back to SAP Commerce Cloud.
6. In Release Quote And Proposal Endpoint, enter the endpoint that sends the quote together with a proposal to SAP Commerce Cloud.
7. Click Save.

Related Information

Prerequisites for SAP Commerce Cloud Integration [page 186]
SAP Commerce Cloud Integration Scenarios [page 189]

1.7.3 SAP Commerce Cloud Integration Scenarios

Depending on where the user journey starts, there are two use case scenarios: users can begin their journey from SAP Commerce Cloud, or from SAP CPQ.

Scenario 1: Sending a request for quotation from SAP Commerce Cloud

In this scenario, SAP Commerce Cloud is the starting point. The user flow is described below.

1. User sends a request for quotation.
   A user (B2B customer) first creates a quote in SAP Commerce Cloud and decides to ask for a discount or a promotional product by entering a comment in the quote header or on the item level. The quote from SAP Commerce Cloud is then sent to the Request for Quotation Web Method over SAP Cloud Platform Integration as an XML payload, and the External Quote Status custom field is set to BUYER_SUBMITTED.
SAP CPQ, a new quote is created. The products, price details and comment on the quote are the same as in SAP Commerce Cloud. The newly created quote is in the Preparing status. Additionally, customers sent from SAP Commerce Cloud will be either created (if their customer ID does not exist) or updated in SAP CPQ as Bill-to customers. SAP CPQ then sends the previously created notification about the request for quotation to sales representatives.

2. Sales representative reviews the request for quotation.
Sales representative receives the notification about the request for quotation and clicks the link in the email. Upon landing on quote from the email, the sales representative first needs to reassign the quote to himself. In addition, the number of comments the user entered is displayed on the Collaboration/Comments button in the top-right corner. After reading the user’s/B2B customer’s comments, the sales representative reviews the request. Additionally, the sales representative decides whether or not to make the required changes (for example, the sales representative decides to apply the requested discount) and notifies the user by replying back to the comments in the Collaboration/Comments section.

3. Sales representative releases the quote.
After applying the changes and replying back to the customer, the sales representative clicks the Release Quote To SAP Commerce action to send the quote back to SAP Commerce Cloud. SAP Commerce Cloud then applies all changes received from SAP CPQ. Additionally, in SAP CPQ, the sales representative can also release the quote together with a proposal. SAP Commerce Cloud only supports proposals in PDF format. In this case, when the sales representative clicks the Generate Documents action, a new page displays, enabling the sales representative to also write a message for the user (B2B customer). In order to release the quote with proposal, the sales representative needs to click the Submit to customer button. Upon clicking the button, the XML payload of the quote, the generated proposal and the message (if the sales representative wrote it in the dedicated text field) are sent to SAP Commerce Cloud.

4. Customer accepts or rejects the quote.
When the user (B2B customer) accepts the quote received from SAP CPQ, SAP Commerce Cloud calls the Perform Cart Action Web Method and changes the quote status to Customer Accepted. In addition, the External Quote Status custom field is set toOrdered. The order is placed in SAP ERP and the SAP ERP sales order ID is saved on the quote in SAP CPQ. Apart from accepting the quote in SAP Commerce Cloud, the user can either edit or reject the quote.
In case the user edits the quote received from SAP CPQ (for example, the user modifies the additional discount) and sends a new request for quotation for the same quote in SAP Commerce Cloud, a new quote revision, together with all data from the request for quotation, will be created in SAP CPQ. The revision will be automatically active. If the user rejects the quote received from SAP CPQ, the Perform Cart Action Web Method is called, and the quote status is changed to Customer Rejected. Also, the External Quote Status custom field is set to CANCELLED. If there are any comments either on the quote header or item level, they are also sent. After rejecting the quote, the user needs to send a new request for quotation from SAP Commerce Cloud, which will create a new quote in SAP CPQ.
1.8 SAP Commerce Cloud Integration for Quote 2.0

The integration between SAP CPQ and SAP Commerce Cloud allows SAP Commerce Cloud customers to negotiate on quote specifics and send a request for quotation to SAP CPQ in one click. They can add subscription and standard physical products and ask for discounts and change of quote terms by leaving comments on the quote and sending it to SAP CPQ.

The integration described in this section works only in the Quote 2.0 engine and it uses SAP Cloud Platform Integration as middleware for communicating data across systems. The corresponding integration for the Quote 1.0 engine is described in SAP Commerce Cloud Integration [page 185].

**Note**
The integration details on the SAP Commerce Cloud side, as well as how-to instructions are available in the SAP Commerce Cloud documentation.

**SAP Commerce Cloud User Scenarios** [page 191]

There are two typical scenarios for users in the SAP CPQ - SAP Commerce Cloud integrated environment. One of them is initiated from SAP Commerce Cloud and the other is initiated by sales representatives in SAP CPQ.

**Setting Up SAP Commerce Cloud Integration for Quote 2.0** [page 193]

There's a predefined list of steps that you need to perform to enable the integration between SAP CPQ and SAP Commerce Cloud for the Quote 2.0 engine.

**Field Mapping and Data Exchange** [page 199]

The SAP Cloud Platform Integration iFlows contain the definition of field mapping between SAP CPQ and SAP Commerce Cloud. This page describes the field mapping and explains the data exchange between the systems.

1.8.1 SAP Commerce Cloud User Scenarios

There are two typical scenarios for users in the SAP CPQ - SAP Commerce Cloud integrated environment. One of them is initiated from SAP Commerce Cloud and the other is initiated by sales representatives in SAP CPQ.

**Note**

SAP CPQ-SAP Commerce Cloud integration is supported only for the Quote 2.0 engine.
Scenario 1 - from SAP Commerce Cloud to SAP CPQ

A customer is on the SAP Commerce Cloud website browsing through products (physical products coming from SAP ERP and subscription products coming from SAP CPQ). When products are added to the quote in SAP Commerce Cloud, the customer submits it to SAP CPQ. If the customer wishes to get benefits for the selected products, or change quote terms, they can add comments on the quote level and item level asking, for example, for a higher discount, a lower price, and so on.

The Submit Quote action sends the SAP CPQ quote as XML payload from SAP Commerce Cloud to SAP CPQ over SAP Cloud Platform Integration. In SAP Cloud Platform Integration, fields on the quote originating from SAP Commerce Cloud are mapped to the fields on the new quote created in SAP CPQ, as defined in the artifacts [page 196]. SAP Cloud Platform Integration uses the POST quote API to create a quote with market, customer details, custom fields, comments, and items. From this point, the quote in SAP Commerce Cloud is locked and can’t be changed. On the other hand, the sales representative in SAP CPQ receives a notification that there’s a new request for quotation that should be reviewed. The quote is assigned to the integration user who made the quote and it’s displayed in the Other Quotes tab. The sales representative should find the quote and reassign it. The quote will then display in the My Quotes tab.

Learn more about field mapping and data exchange between SAP CPQ and SAP Commerce Cloud [page 199].

The sales representative opens the SAP CPQ quote, reviews the comments from the customer, and eventually makes the requested changes. In this point, the quote may be sent for approval if the changes exceed the limits set by the administrator. Additionally, the sales representative may generate a document to be sent to SAP Commerce Cloud with the quote.

i Note
SAP Commerce Cloud only accepts documents in the PDF format.

In the end of the usual document generation procedure in SAP CPQ, for this integration only, a Send to Commerce modal appears, in which the sales representative can enter additional comments, if needed. Afterwards, the sales representative sends the quote back to SAP Commerce Cloud via SAP Cloud Platform Integration that works as the middleware for syncing data. Depending on whether the quote contains a generated proposal or not, the sales representative clicks the Release Quote To SAP Commerce or the Release Quote with Proposal To SAP Commerce action in SAP CPQ.

At this point of the user journey, there may be as many cycles of quote negotiation as needed. Along the process, the quote status changes accordingly [page 198] and data is synchronized as defined in the artifacts. Every time new data set is sent to SAP CPQ, the system creates a new revision of the same quote, while in SAP Commerce Cloud with each new update, there’s a new quote in the system. Whenever a new quote revision is created, the sales representative receives the SAP Commerce Cloud Request for New Quote Revision notification.

Once the negotiation phase is successfully completed, the SAP Commerce Cloud customer accepts the quote and the order for physical products is sent to SAP ERP and the order for subscription products is sent to SAP Subscription Billing.

Scenario 2 - from SAP CPQ to SAP Commerce Cloud

The flow in this scenario is the same as in the Scenario 1, except that in this scenario the user journey starts in SAP CPQ. The sales representative first creates a quote based on the data received from the customer offline.
(for example, in a telephone call). After adding items to the quote, the sales representative may add **general and item-specific comments** to the quote, if needed. When the quote is ready, user clicks the **Release Quote to SAP Commerce** action.

**Note**

In this scenario, it is not possible to send the quote with the generated document proposal to SAP Commerce Cloud from SAP CPQ the first time the quote is created. In this scenario, this is currently possible only in the following cycles of negotiation, when the SAP Commerce Cloud returns the quote to SAP CPQ and a new revision is created. In **Scenario 1**, generated documents can be sent to SAP Commerce Cloud in any cycle of negotiation.

The quote is now created in SAP Commerce Cloud and the quote number corresponds to the **Quoteld** of the quote in SAP CPQ that is sent to SAP Commerce Cloud in the payload.

In SAP Commerce Cloud, the customer can accept the quote and proceed to checkout, edit the quote, or cancel it. Whatever the option the customer chooses, the status of the quote is changed accordingly in SAP CPQ. Additionally, if the customer edits the quote, a new revision of the quote is created in SAP CPQ with the same name as the quote version in SAP Commerce Cloud.

**Note**

All revisions of a quote are tied to the same quote composite number, but each revision has a different **Quoteld**. Consequently, whenever a new revision of the quote is sent to SAP Commerce Cloud, the system there creates a new quote with a different quote number.

**Related Information**

[Setting Up SAP Commerce Cloud Integration for Quote 2.0](#) [page 193]
[Field Mapping and Data Exchange](#) [page 199]
Quote 2.0
SAP Commerce Cloud Documentation

### 1.8.2 Setting Up SAP Commerce Cloud Integration for Quote 2.0

There’s a predefined list of steps that you need to perform to enable the integration between SAP CPQ and SAP Commerce Cloud for the Quote 2.0 engine.

**Setup Steps Overview**

Below is an overall list of steps that need to be performed for the integration between SAP CPQ and SAP Commerce Cloud to work in the Quote 2.0 engine.
1. Provide SAP CPQ and SAP Commerce Cloud tenants that can be integrated.
2. Make sure that Quote 2.0 is enabled on the SAP CPQ tenant.
3. The application parameter *Allowed origins for the CORS filter* must be configured in the way explained in Application Parameters.
4. Turn on the integration. [page 194]
5. Set up the common settings [page 195].
6. Set Up Subscription to Events [page 196]
7. Enable integration with SAP Subscription Billing. [page 106]
8. Set up the workflow in SAP CPQ [page 198].
9. Change the *iFlows created for this integration* [page 196] in SAP Cloud Platform Integration if your business model requires different field mapping [page 199].

**Enable SAP Commerce Cloud Integration for Quote 2.0** [page 194]

Turning on the integration in the Providers in SAP CPQ is the first step toward an integrated environment with SAP Commerce Cloud.

**Common Settings** [page 195]

In the Common Settings, you need to define the SAP Cloud Platform Integration endpoints to enable the quote payload to reach SAP Commerce Cloud via SAP Cloud Platform Integration.

**Set Up Subscription to Events** [page 196]

To enable SAP CPQ to send information to SAP Subscription Billing and SAP Commerce Cloud over SAP Cloud Platform Integration, you need to create subscriptions to events, which send notifications from SAP CPQ to SAP Cloud Platform Integration whenever certain events are triggered.

**Set Up Workflow in SAP CPQ** [page 198]

The Workflow in SAP CPQ needs to be set up in a way that allows streamlined user journey to and from SAP Commerce Cloud.

**Enable SAP Commerce Cloud Integration for Quote 2.0**

Turning on the integration in the Providers in SAP CPQ is the first step toward an integrated environment with SAP Commerce Cloud.

1. Go to Setup Providers Providers SAP.
2. Available SAP providers display.
3. Click SAP Cloud Platform.
4. Open the SAP Commerce Cloud Settings tab.

**Note**

If you don’t have this tab available, contact the SAP Support team to enable this for you.

4. Turn on the Connect SAP Commerce Cloud toggle.

The integration is now enabled.

You now need to perform one more step as explained below.
Additional Step: Manage Notification Recipients

Enabling the integration with SAP Commerce Cloud automatically creates two email notifications. SAP Commerce Cloud Request for New Quotation is sent to sales representatives in SAP CPQ when the customer creates a quote in SAP Commerce Cloud for the first time and sends it to SAP CPQ. The SAP Commerce Cloud Request for New Quote Revision is sent afterwards, each time the quote has been updated on the SAP Commerce Cloud side and sent back to SAP CPQ.

Furthermore, the email list SAP Commerce Cloud Request for New Quotation is created and you need to add the recipients for the SAP Commerce Cloud Request for New Quotation notification. Go to Workflow/Approval Email Lists open SAP Commerce Cloud Request for New Quotation, and add users.

Related Information

Common Settings [page 195]
Set Up Subscription to Events [page 196]
Set Up Workflow in SAP CPQ [page 198]
Field Mapping and Data Exchange [page 199]
SAP Commerce Cloud Documentation

1.8.2.2 Common Settings

In the Common Settings, you need to define the SAP Cloud Platform Integration endpoints to enable the quote payload to reach SAP Commerce Cloud via SAP Cloud Platform Integration.

1. Go to Providers Providers SAP SAP Cloud Platform .
2. Turn on the Enable integration toggle.
3. In Hana Cloud Platform Integration - Authentication details, fill out the following fields:
   ○ Authentication mode - BasicAuth
   ○ Integration username - the username of the integration user configured in the SAP Cloud Platform Integration tenant
   ○ Integration password - password configured for the integration user
4. In General settings, define the following:
   ○ Send customers along with quote payload - when the toggle switch is enabled, SAP CPQ sends customer details such as bill-to, ship-to and end user details, together with the quote data to the SAP system. The quote data is sent in XML format.
   ○ Dates in quote standard fields (e.g. Date Created, Date Modified...) will be converted into this time zone - You can select either UserZone or UtcZone.
5. In SAP Cloud Platform Integration Service API Details, enter the following:
   ○ REST API base URL address - the base REST API URL address of the SAP Cloud Platform Integration – SAP CPQ specific package. The value is delivered by the administrator and should not be changed. In addition, the value identifies a specific SAP Cloud Platform Integration tenant.
○ **Endpoint that gets SAP object data details** - a relative endpoint address of the SAP Cloud Platform Integration artifact that gets an object from the SAP system over SAP Cloud Platform Integration. The call acts as a proxy to SAP Data services and can query any publicly visible object from SAP Commerce Cloud.

○ **Endpoint that processes posted data in xml format** - a relative endpoint of the SAP Cloud Platform Integration artifact that processes the posted data in the XML format.

○ **Endpoint that sends CPQ quote to SAP systems** - a relative endpoint of the SAP Cloud Platform Integration artifact that sends the SAP CPQ quote to SAP systems.

### Related Information

- Set Up Subscription to Events [page 196]
- Set Up Workflow in SAP CPQ [page 198]
- Field Mapping and Data Exchange [page 199]
- SAP Commerce Cloud User Scenarios [page 191]
- SAP Commerce Cloud Documentation

### 1.8.2.3 Set Up Subscription to Events

To enable SAP CPQ to send information to SAP Subscription Billing and SAP Commerce Cloud over SAP Cloud Platform Integration, you need to create subscriptions to events, which send notifications from SAP CPQ to SAP Cloud Platform Integration whenever certain events are triggered.

#### i Note

You can learn how to create subscriptions to events in Subscription to Events [page 201].

The SAP Cloud Platform Integration package provided for this integration contains the following IFlow packages:

- **SAP Commerce Integration with SAP CPQ Quote 2.0**
- **SAP Commerce Cloud Integration with SAP CPQ and SAP Subscription Billing**

In Subscription to Events, you need to create subscriptions to events for each of the IFlows. Listed below are the four subscriptions you should create for this integration.

#### Quote Release to Commerce

- **Name**: Quote Release to Commerce
- **Description**: this field is optional.
- **Webhook URL (required)**: enter the URL for the **SAP Commerce Integration with SAP CPQ Quote 2.0** IFlow provided in the SAP Cloud Platform Integration package.
- **Webhook HTTP method (required)**: POST
- **Event name (required)**: QuoteReleaseToCommerce
- **Authentication type (required)**: Basic
• **Username (required):** this should match the information from SAP Cloud Platform Integration.
• **Password (required):** this should match the information from SAP Cloud Platform Integration.

### Release Quote with Proposal

• **Name:** Release Quote with Proposal
• **Description:** this field is optional.
• **Webhook URL (required):** enter the URL for the SAP Commerce Integration with SAP CPQ Quote 2.0 IFlow provided in the SAP Cloud Platform Integration package.
• **Webhook HTTP method (required):** POST
• **Event name (required):** QuoteAndProposalReleaseToCommerce
• **Authentication type (required):** Basic
• **Username (required):** this should match the information from SAP Cloud Platform Integration.
• **Password (required):** this should match the information from SAP Cloud Platform Integration.

### Create Product

• **Name:** Commerce-Create Product
• **Use for Subscription Billing products:** enable the toggle.
• **Description:** this field is optional.
• **Webhook URL (required):** enter the URL for the SAP Commerce Cloud Integration with SAP CPQ and SAP Subscription Billing IFlow provided in the SAP Cloud Platform Integration package.
• **Webhook HTTP method (required):** POST
• **Event name (required):** ProductAdded
• **Authentication type (required):** Basic
• **Username (required):** this should match the information from SAP Cloud Platform Integration.
• **Password (required):** this should match the information from SAP Cloud Platform Integration.

### Update Product

• **Name:** Commerce-Update Product
• **Use for Subscription Billing products:** enable the toggle.
• **Description:** this field is optional.
• **Webhook URL (required):** enter the URL for the SAP Commerce Cloud Integration with SAP CPQ and SAP Subscription Billing IFlow provided in the SAP Cloud Platform Integration package.
• **Webhook HTTP method (required):** POST
• **Event name (required):** ProductUpdated
• **Authentication type (required):** Basic
• **Username (required):** this should match the information from SAP Cloud Platform Integration.
• **Password (required):** this should match the information from SAP Cloud Platform Integration.

### Related Information

Set Up Workflow in SAP CPQ [page 198]
Field Mapping and Data Exchange [page 199]
Common Settings [page 195]
SAP Commerce Cloud User Scenarios [page 191]
1.8.2.4 Set Up Workflow in SAP CPQ

The Workflow in SAP CPQ needs to be set up in a way that allows streamlined user journey to and from SAP Commerce Cloud.

Go to Setup > Workflow/Approvals > Workflow, and set the following:

**Note**
Have in mind that the names of actions and the quote statuses may have different naming from what is stated here, as these can be configured in SAP CPQ per client’s specific business model. Also, some statuses are added automatically and some actions are automatically set up to display in the right status. Others need to be set up manually.

**Added Automatically**
- **Released to Commerce** status is automatically created when the integration is enabled.
- **Accept Proposal** action is set up to lead from the Released to Commerce status to Customer Accepted.
- **Reject proposal** action is set up to lead from the Released to Commerce status to Customer Rejected.
- Action **New Active Revision** is automatically set up to lead from the Released to Commerce status to Preparing and the SAP Commerce Cloud Request for New Quote Revision notification is attached to it.

**To be Added Manually by the Administrator**
- Action **Release Quote To SAP Commerce** to lead to the Released to Commerce status after the quote is released to SAP Commerce Cloud.
- Action **Release Quote with Proposal To SAP Commerce** to lead to the Released to Commerce status after the quote with a generated proposal is released to SAP Commerce Cloud.
- Quote in the Released to Commerce status should be locked for any changes. To lock the quote for changes, disable the Edit action and enable the View action in Workflow.

**Related Information**

Field Mapping and Data Exchange [page 199]
SAP Commerce Cloud User Scenarios [page 191]
Workflow-Approvals
SAP Commerce Cloud Documentation
1.8.3 Field Mapping and Data Exchange

The SAP Cloud Platform Integration iFlows contain the definition of field mapping between SAP CPQ and SAP Commerce Cloud. This page describes the field mapping and explains the data exchange between the systems.

Field Mapping

<table>
<thead>
<tr>
<th>SAP CPQ</th>
<th>SAP Commerce Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cart Comment</td>
<td>Description</td>
</tr>
<tr>
<td>Collaboration/Comments</td>
<td>Comments in the quote header</td>
</tr>
<tr>
<td>Global comment Item-level comment in Collaboration/Comments</td>
<td>Comment on item level</td>
</tr>
<tr>
<td>Revision Name</td>
<td>Version</td>
</tr>
<tr>
<td>External Id</td>
<td>Quote ID</td>
</tr>
<tr>
<td>Sales Order Id</td>
<td>The ID of the order sent from SAP Commerce Cloud to SAP ERP. This field is in the quote header and it’s empty until the customer in SAP Commerce Cloud accepts the quote and the order is sent to SAP ERP. Sales order ID is calculated by adding 3 on the Order Number in SAP Commerce Cloud.</td>
</tr>
</tbody>
</table>

Discount Percent | Discount

Data Exchange

Integration User

The integration user’s purpose is to call SAP CPQ APIs in SAP Cloud Platform Integration. This user needs to be defined in both SAP CPQ and SAP Cloud Platform Integration. Its username and password need to be added in the subscription to events [page 196] defined in SAP CPQ.

Customers

Customers are synchronized in both SAP Commerce Cloud and SAP CPQ from SAP ERP. However, when the user journey starts in Global comment in SAP Commerce Cloud and there isn’t yet a customer in SAP CPQ and SAP ERP, it’s created in both systems.

i Note

Since SAP ERP integration is still not supported on Quote 2.0, all communication between SAP CPQ and SAP ERPIs managed through SAP Cloud Platform Integration.

Market

Markets aren’t automatically synchronized between systems so the administrator in SAP CPQ should create it manually. The Market Code in SAP CPQ corresponds to Sales Organization ID in SAP Commerce Cloud.
**Items**

SAP CPQ - SAP Commerce Cloud integration for Quote 2.0 supports subscription products and physical products (coming from SAP ERP via SAP Cloud Platform Integration). All subscription products need to be created first in SAP CPQ since SAP CPQ is the master for subscription billing products. After a subscription product is created, an event is triggered and product details are sent to SAP Subscription Billing and SAP Commerce Cloud where the same product is created.

**Custom Fields**

There are no custom fields that are necessary in the integration model. However, in SAP Cloud Platform Integration, you can map the custom fields in SAP CPQ with the fields in SAP Commerce Cloud, as needed. The values of custom fields in SAP CPQ are updated accordingly via the API.

**i Note**

Currently, it isn’t possible to manage the quantity of SAP Subscription Billing products in SAP Commerce Cloud via the QTY field. Instead, the customer and the sales representative should manually enter as many SAP Subscription Billing products as needed.

**Related Information**

SAP Commerce Cloud User Scenarios [page 191]
Setting Up SAP Commerce Cloud Integration for Quote 2.0 [page 193]
SAP Commerce Cloud Documentation
2 Subscription to Events

You can subscribe to domain events by establishing a connection between the application and an external URL to which notifications will be sent whenever the selected event occurs on products or in quotes.

The purpose of this feature is notifying clients, or whoever has access to the provided URL, about specific actions performed on products or in quotes. It can be accessed in Setup Providers Subscription to Events.

Available Subscription Events

There are several domain events that can be triggered in SAP CPQ.

Product Subscription Events

- **ProductCreated** - triggered when a product is created in SAP CPQ.
- **ProductUpdated** - triggered when a product is updated in SAP CPQ.
- **ProductDeleted** - triggered when a product is deleted from SAP CPQ.

These events are triggered when a product is added, updated or removed via API calls for products, in Setup sections: Bulk Import/Export, Product Import, Deploy/Send Changes, Products, Products 2.0 and the Manage Simple Products option in the User Menu.

Quote Subscription Events

- **QuoteProvisioned** - triggered when the Place Order action is executed in Quote 2.0.
- **OnQuoteDeleted** - triggered when the quote is deleted from SAP CPQ.

Attribute Subscription Events

- **AttributeCreated** - triggered when an attribute is created in SAP CPQ.
- **AttributeUpdated** - triggered when an attribute is updated in SAP CPQ.
- **AttributeDeleted** - triggered when an attribute is deleted in SAP CPQ.

There are also subscription events that are available only for the Quote 2.0 when the integration with SAP Commerce Cloud is enabled [page 196].

Related Information

Adding Subscriptions to Events [page 108]
Verify Successfully Notified Logs in SAP CPQ Log [page 203]
2.1 Set up Subscription to Events

Once you are in Setup > Providers > Subscription to Events, you can see the list of existing subscriptions to events. They are divided into several columns:

- **Edit** - this column contains the Edit and Delete buttons.
- **Name** - this column contains descriptions for each of the listed subscriptions.
- **Description** - this column contains descriptions for each of the listed subscriptions.
- **Webhook URL** - this column contains URLs to which the notifications are sent once the events are triggered.
- **Event name** - this column contains names of the events to which you subscribe.

In order to create subscriptions to events, that is, to establish a connection between a triggered event and an external URL, follow the steps provided below. Once you create a subscription, notifications will be sent to the specified external URL whenever the events occur:

1. Go to Setup > Providers > Subscription to Events.
   The page in which the existing domain events are listed displays.
2. Click Add new Subscription to Event.
   The page for adding a new subscription to events displays, containing the following options:

   | Name (required) | Needs to contain the name of the subscription. |
   | Use for Subscription Billing products | Should be ticked in order to subscribe to events tied to products which use the subscription billing pricing mechanism (relevant for product domain events only) |
   | Description | Can contain a subscription description. |
   | Webhook URL (required) | Needs to contain the URL to which you want notifications to be sent whenever the event is triggered. |
   | Webhook HTTP method (required) | Needs to contain a HTTP method for sending requests whenever the event is triggered. |
   | Event name (required) | Needs to contain the name of the event you wish to subscribe to. |
   | Authentication type (required) | You can choose between Basic and OAuth. |
   | Username | Displays if Basic authentication is selected. The field needs to contain your username. |
   | Password | Displays if Basic authentication is selected. The field needs to contain your password. |
   | Client ID (required) | This field needs to contain the client ID. |
   | Client Secret (required) | Client secret should be entered here. |

3. Click Save if you wish to save the subscription and remain on the page.
4. Click Save & Go Back if you wish to go back to the page with listed domain events.

Once you are on the Subscription to Events page, you can also edit or delete subscriptions to events by clicking Edit or Delete buttons.
Once the new subscription to events is created, notifications will be sent to the provided Webhook URL seven times at most (by default), before the unsuccessful attempt is recorded in the Log. However, no further attempts at sending notifications to the specified Webhook URL will be made as soon as the first successful attempt occurs.

You can filter the existing subscriptions to events by writing their names, webhooks or related actions in the intended textbox located above the listed subscriptions. The corresponding subscriptions will be listed as soon as you finish typing one of these pieces of information.

You can select which columns you wish to be displayed in the Subscription to Events page in Display Settings.

### 2.2 Verify Successfully Notified Logs in SAP CPQ Log

By checking the following pipeline, you can verify that successfully notified logs exist in the SAP CPQ log.

1. The Product Added, Product Updated, Product Deleted or any other event is triggered.
2. In the event log there should be a message Sent message to producer with Key: . The key should be {cpq url}:{domain}:{eventKey of triggered event}.
3. After that the following sequence should take place:
   1. A message from Kafka was received in Notifier MS.
   2. Notification saving was started.
   3. Notifier MS received a message and attempted to match with the appropriate webhook.
   4. Notification was saved.
   5. Match was found.
4. Successfully notified 1/1 - the Notifier MS successfully sent a notification about a matched webhook. Alternatively, the Notifier MS failed to notify because the webhook (target URL) is not available or reachable, or the authentication is wrong. In case notifying fails, check your target URL and authentication.

**Note**

Notifier can notify about more than 1 event in a single batch if more events of same type are received within short time period.

Related Information

Subscription to Events [page 201]
### Federated Single Sign-On (FSSO)

**Single sign-on (SSO)** is a session/user authentication process that permits users to enter a single name and password to access multiple applications. While SSO uses a single login (username/password) to access all applications within the same organization, **federated SSO (FSSO)** goes a step further and extends SSO across enterprises.

In other words, FSSO allows access to multiple systems across different organizations, benefitting both users and organizations.

---

**Prerequisites**

To be able to successfully use the FSSO in SAP CPQ, you must complete these steps:

1. Contact the SAP CPQ Support team to enable access to the Federation Settings [page 206] section of the Setup.
   
   In case multiple SAP CPQ domains on a single SAP CPQ environment are using the same identity provider, request federation support for your environment from your SAP CPQ contact person by providing a list of domains for which the federation dashboard needs to be enabled.

2. Configure the settings in the Federation Settings [page 206] section and set up an operational identity provider.
   
   You can do this on your own.

3. Create local SAP CPQ users.
   
   The toggle switch **Must Change Password at Next Login** in [Setup] ➔ [Users] ➔ [Create New User/Edit Password Settings] must be disabled for all FSSO users.

4. Form the federation URL [page 210] for SAP CPQ SP-initiated FSSO for your environment.

---

### SAP CPQ Federation Technical Overview

The SAP CPQ federated authentication uses the Security Assertion Markup Language (SAML) 2.0 protocol, allowing you to exchange authentication and authorization data between cross-domain applications. Consequently, this allows you to sign onto a remote IdP and to access the SAP CPQ application. The federated authentication using SAML can be enabled on request for your organization. Information about SAML 2.0 protocol details is available in [Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0](https://docs.oasis-open.org/security/saml/v2.0/saml acronym). For remote IdPs, SAP CPQ acts as a Service Provider (SP). SAP CPQ never works as an IdP. In Microsoft terminology, Relaying Party (RP) stands for Service Provider.

SAP CPQ supports two major SAML 2.0 profiles:
Federated Single Sign On (FSSO) - signs on users on all systems in the federation trust.

Federated Single Sign Out (FSLO) - signs out users from all systems in the federation trust.

More details about SAML 2.0 profiles are available in Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0.

SAP CPQ does not support the Web Service Federation protocol (WS-Federation/Ws-Fed).

Benefits of Single Sign-On

Setting up FSSO provides you with the following advantages:

- Improved user experience - With SSO, users authenticate only once at a single point and enjoy a seamless experience across multiple domains. There is no need to remember separate credentials for each cross-domain because users retain only one set of credentials. This means they can securely move between services with no interruptions and without having to enter their credentials upon entering each new domain.

- Saving time - When working with cross-domain applications, it can take up to 30 seconds to sign on to a web application or even longer if users mistype their username or password and have to re-enter it. SSO solves this problem by having the user enter credentials only once on the identity provider (IdP) side. By saving time, users also boost their productivity.

- Enhanced security - User credentials are provided directly on the central SSO server, not on the actual service the user is trying to access. As a result, the credentials cannot be cached by the service. The central authentication point, the SSO service, reduces the possibility of phishing.

- Simplified password management - Reduces administrative overhead in resetting forgotten passwords over multiple platforms and applications.

Federated Single Sign-On

SAP CPQ supports the SP- and IdP-initiated single sign-on. SP FSSO is initiated by the SAP CPQ application. When SP FSSO is utilized, SAP CPQ, as a SaaS application, exposes the new URL address specific to your tenant environment. On the other hand, IdP-initiated FSSO begins from IdP and, following authentication, lands the user on the SAP CPQ default page.

- Redirect binding is preferred for the FSSO profile (authentication flow): `-urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect`

- Artifact binding is not supported at this time: `-urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact`

Binding details are specified in the SP metadata that is sent to you for the specific SAP CPQ environment.
Federated Single Sign-Out


SAP CPQ supports local and federated single sign-out scenarios. For the post sign-out, we support default and custom processing. Custom processing covers response actions received after sign-out (for example, redirecting to specific URL or calling a publicly visible API).

**i Note**

SAP CPQ can process two types of XML metadata files:

- files that describe a single entity, with the root element `<EntityDescriptor>`
- files that describe multiple entities, with the root element `<EntitiesDescriptor>`, containing a sequence of `<EntityDescriptor>` elements.

Furthermore, the federation is able to process a metadata file that contains either one or more federation signing and/or encryption certificates - that is, the file may include the certificate that is currently valid and one or more certificates that become valid after the currently valid certificate expires.

### 3.1 Federation Settings

The **Federation Settings** subsection of Setup > Providers allows you to independently download and upload metadata documents and to add, select, and configure identity providers.

That way, you can establish the Federated Single Sign-On (FSSO) [page 204], which allows you to securely access multiple applications via a single set of credentials.

To enable access to the Federation Settings page in the Setup, contact the SAP CPQ Support team.

**i Note**

For a list of common issues related to setting up the FSSO in SAP CPQ, see Federation Troubleshooting.

### Service Provider Settings

Here, you can find the URL of the environment you’re currently on. The URL listed here is the one that needs to be used for forming the federation URL [page 210] after configuring the settings in this section.

### Identity Providers Settings

Here, you can add, select, or configure your identity provider.
• **Select identity provider** - select an existing identity provider from the dropdown menu. In order to appear in this menu, existing identity providers need to be enabled by the SAP CPQ Support team for each domain individually.

• **Add new identity provider** - click to add a new identity provider. Once you fill in the fields, make sure to save and activate your settings. Once you create a new identity provider, it’s visible only in the domain where it was created. To make it visible on other domains, you need to contact the SAP CPQ Support team.

You can configure your identity provider via the following fields:

• **Download SAP CPQ metadata with selected certificates** – download the service provider XML metadata file with the signing and encryption certificates that are selected for that specific selected identity provider. This button becomes available after an identity provider is selected or added.

• **Name (required)** - enter the identity provider application unique name.

• **Metadata location (required)** - this field contains the relative path to the identity provider metadata XML file that was uploaded. The metadata file is typically stored in App_Data folder.
  
  ○ **Download metadata document** - click to download the identity provider XML data file. You can download the IdP file if you need to compare and check if there have been any updates to it. If nothing is updated, there’s no need to reupload the file - instead, you only need to update some settings (for example, a routing or an attribute).
  
  ○ **Upload metadata document** - click to upload the identity provider XML data file.

  **i Note**
  
  When you're uploading a metadata document, you need to add at least one valid signing certificate to the IDPSSODescriptor node before you can upload it successfully.

• **Entity ID** - the unique entity ID of the entity provider. This field is automatically populated with the information from the metadata document.

• **ReadUIDFrom** - determine where the Uid is read from. When the user logs in using an IdP and lands to SAP CPQ, the ReadUIDFrom field (unique identifier) is used for identifying the user. The Uid can be a username, an email, a custom ID, or any other value. If the value of this field is different from the username, you need to enter that value in the **Federation ID** in >> Setup >> Users >> Users >> Edit >> General >>. If the value of this field is the same as the username, no mapping is necessary for the individual user.
  
  ○ **Assertion** - select this to make sure that the Uid is read from the list of assertions in the SAML. If there’s no Uid in the SAML, or if you want the Uid to be read from a different source, you need to select **Assertion** and then proceed to configure the **Attribute Mappings**.
  
  ○ **NameId** - select this to make sure that the User ID is a fixed value read from the NameId field in the SAML.
  
  If the incorrect option is selected here, an error appears when the user tries to sign in.

• **SignOut** - select the sign-out method to be used with this identity provider.
  
  ○ **Federated** - after signing out, you’re signed out of every application within the federation.
  
  ○ **Local** - after signing out, you’re only signed out of the application you are in.
  
  ○ **Custom** - specify an address to which the sign-out request is sent (used if, for example, a script needs to be executed when the user signs out) by selecting this option and setting a federation parameter with a **SignOutAddress** value.

• **PostSignOut** - select the post sign-out method to be used with this identity provider. This method is executed upon the sign-out response.
  
  ○ **Default** - no method is executed upon signing out.
  
  ○ **Custom** - specify an address to which you’re redirected or to which an additional request is sent after signing out by setting a federation parameter with a **RedirectAddress** value. For example, once the user
signs out locally, you can make a message appear, stating that the sign-out was local and that the session is still running on the IdP. In that case, the static URL hosting that message is outside SAP CPQ, and needs to be mapped via this field and the corresponding parameter. A custom post sign-out can be combined with a custom sign-out method.

- **Description** - enter a description for the identity provider.
- **Signing certificate (required)** - select the signing certificate you wish to use with your identity provider. The dropdown menu lists all the certificates that your current environment supports, but only the selected certificate is in use. You can use a different signing and encryption certificate. The certificate selected here is downloaded in the service provider XML metadata file when you click Download SAP CPQ metadata with selected certificates.
- **Encryption certificate (required)** - select the encryption certificate you wish to use with your identity provider. The dropdown menu lists all the certificates that your current environment supports, but only the selected certificate is in use. You can use a different signing and encryption certificate. The certificate selected here is downloaded in the service provider XML metadata file when you click Download SAP CPQ metadata with selected certificates.
- **Sign In Request Binding (required)** - select the sign-in request binding protocol.
- **Sign Out Request Binding** - select the sign-out request/response binding protocol.

You can further configure your identity provider settings via the following toggle switches:

- **Ignore assertion attributes validation** - define whether or not received assertions should be validated against the expected attribute list.
- **Should the artifact resolve be signed** - define whether or not post responses should be signed.
- **Should the sign out request be signed** - define whether or not the sign-out request should be signed.
- **Should the sign out response be signed** - define whether or not the sign-out response should be signed.

### Federation Parameters

Here, you can edit the list of the defined federation parameters for the selected identity provider. Used when a custom sign-out or custom post sign-out is used.

**Note**

When adding new federation parameters, make sure that the parameter name is in one of the following recommended formats:

- `{IdentityProviderName} + {"RedirectAddress", "SignOutAddress"}`
- `{EntityIdForSAMLBearer_" + {IssuerLink}`

For example, if your IdP name is Adfs and you want to be redirected to [www.google.com](http://www.google.com) when you sign out, you need to select **Custom** under **PostSignOut**, and add a parameter here with the name `AdfsRedirectAddress` and the value [www.google.com](http://www.google.com).

### Routings

In this section of the page, you can define the routings with the relative URL segments used to access the application via the federation protocol. The URL segment is used for SP-initiated FSSO. The routing name and
the URL should be unique. It’s necessary to define and enable at least one routing in this section for the federated single sign-on to work. Moreover, here you can see a list of all the routings that exist for the IdP.

You can create multiple routings to be used for accessing the same identity provider. The system generates every new routing based on your current tenant and the ordinal number of the newly created routing in the list of routings. To add a new routing, follow the procedure below.

1. Click **Edit** under **Routings**. **Save** and **Add** buttons display.
2. Click **Add**.
   A new routing is added.
3. In the **Sign on relay state** dropdown menu, select the tenant. Only one tenant is displayed, unless multiple tenants on a single environment have a single identity provider, in which case you should select the tenant for which you’re setting up the routing. Alternatively, if you leave this field blank, the default tenant is used.
4. Click the **Enabled** toggle switch to enable the routing. Alternatively, you can disable a routing via the **Enabled** toggle switch.
5. Click **Save** to save the changes made to routings. The routing with the defined relative URL segment is now enabled and can be used for accessing the application. You can now proceed to form the federation URL [page 210].

**Attribute mappings**

Here, you can edit the list of attribute mappings. Each mapping is processed with an authentication response received from the identity provider. Attribute mappings are used to override existing values or add new values to the ones from the metadata XML file. Mapping can change an existing attribute’s name or value or add a new attribute with a constant value (such as application domain).

To make sure that the Uid is read from a source that is neither the **NameId** field nor the list of assertions in the SAML, after selecting **ReadUIDFrom** **Assertion**, follow the steps.

1. Under **Attribute Mappings**, click **Add**. A new row appears.
2. Under **Action (required)**, select **OverrideName**.
3. Under **Name (required)**, enter the name of the field in the SAML that you want to map to a different field (for example: email).
4. Under **Name Override**, enter the name of the field to which you want to map the field you named in step 2 (for example: uid).
5. Select **Enable**.
6. Click **Save**. The **uid** field now receives data from the **email** field in the SAML. As a result, when a user logs in, SAP CPQ now tries to map that user using the email sent from the IdP. If the user’s username in SAP CPQ is the same as the email, no mapping is necessary. However, if the username is different from the email, it’s necessary to enter the email in the **Federation ID** field in **Setup > Users > Users > Edit > General**.
Save and Activate

Once you’ve configured the settings of the identity provider, follow the instructions to make your settings active.

1. Click **Save** in the bottom-right corner of the page.
   The settings are now saved, although they’re still not active.
   Alternatively, click **Delete** in the bottom-right corner of the page to delete the selected identity provider.
2. Click **Activate** to make the settings active.
   Clicking **Activate** initiates a countdown of 300 seconds (by default). Once the countdown is over, the federation settings become active and can be used.
   Alternatively, click **Deactivate** to initiate a 300-second countdown after which the selected identity provider is deactivated.

**i Note**

To set a specific duration for the activation and deactivation countdown, contact the SAP CPQ Support team.

Related Information

Federation Troubleshooting
Federated Single Sign-On (FSSO) [page 204]
Forming the FSSO URL [page 210]

### 3.2 Forming the FSSO URL

For each identity provider, SAP CPQ generates a specific SP-initiated URL address.

It’s created according to the following pattern:

- SAP CPQ domain address + fed keyword + IdP-specific URL segment
Consider the following example:

- SAP CPQ domain address: https://www.rs.webcomcpq.com. The root address of the domain can be seen in Setup > Providers > Federation Settings under Service Provider Settings.
- IdP-specific URL segment: test. The URL segment is specified for each routing in Setup > Providers > Federation Settings > Routings > Url (required). The IdP-specific URL segment may be the same as the tenant (domain) name, but this isn’t always the case.

The resulting FSSO URL address: https://www.rs.webcomcpq.com/fed/test/. Note that the root domain URL takes you to the local login screen.

Related Information

Federated Single Sign-On (FSSO) [page 204]
Federation Settings [page 206]

3.3 User Mappings

When using the federation authentication, you must also create local SAP CPQ users, because each federated user is mapped to the local user during the authentication process.

Local users are mandatory, because they’re integrated with the local application and the database schema. Following the central authentication process, the central user identifier (UID/UPN) is delivered from the IdP over the SAML protocol.

**Note**

The federated user is mapped to the local user based on the username. Therefore, the same username should exist in SAP CPQ. However, when mapping the central user to a local user with a different username, the FederationId user field needs to be used. The federation module first tries to use the FederationId to map the local user, and then the local username.

When running as a new federation tenant, the best practice is to import all users from your remote LDAP/Active Directory into SAP CPQ over the user import functionality. Users should be imported with the same usernames. SAP CPQ supports the SCIM API as an auto-provisioning mechanism. Information about the SCIM API and its limitations is available in SCIM API. Regardless of whether we use federation, the local users’ passwords are always required when adding/importing users. Even though passwords are required, they aren’t used (users aren’t required to type them anywhere in SAP CPQ if they’re coming to SAP CPQ from the Federated landing page). The recommendation is to put a random secret password into every user’s field and not share this password with other users.
3.4 Items Affected by Federation

The items listed here behave differently when federation is enabled.

NewQuote Web Method

When federation is enabled, the New Quote from Salesforce Web Method is replaced by its counterparts that are explained below.

NewQuoteForFederation web method

The NewQuoteForFederation has the same parameters as the NewQuote web method, with the addition of the identity provider name (idpName). The identity provider name is a unique name of the identity provider in SAML terminology to which SAP CPQ is connected.

The method’s result link contains a federation URL segment. Here is a sample federation URL of a created quote: https://www.webcomcpq.com/fed/test/Login.aspx?quote=CECDCCD2CDC8C8CB

NewQuoteForFederationRouting web method

Another method used when federation is enabled is NewQuoteForFederationRouting. This method is used to create a quote when federation is set up for two or more client’s tenants at the same SAP CPQ instance. In this scenario, a route is defined for each tenant and must be provided as a parameter idpRoutingName for the NewQuoteForFederationRouting web method.

The method’s result link contains a federation URL segment. Here is a sample federation URL of a created quote: https://www.webcomcpq.com/fed/test2/Login.aspx?quote=CECDCCD2CDC8C8CB

Contact the SAP CPQ Support team to get the appropriate values for the parameters idpName and idpRoutingName.

Notification URLs

The PDA approval link and the quote link that SAP CPQ sends within email notifications have their federation counterparts. Depending on the user’s authentication (local SAP CPQ or remote IdP), the system dynamically determines whether the tags return a regular or federated (FSSO) link.

If the user is signed on directly to SAP CPQ and generates a notification that contains these tags, the system sends a non-federated URL to the approver. The approver then has to access SAP CPQ directly. On the
contrary, if the tags are generated by a user who signed on via federation, the approver receives federated links. In that case, the approver needs to sign on via the federation.

Listed here are example links returned via quote link notification tags and PDA approval notification tags for each described scenario.

**Quote link notification tag: `<*QUOTELINK*>`**
- Classic environment result link: https://www.rs.webcomcpq.com/Login.aspx?quote=CAC7C7D2C8
- Federation environment result link: https://www.rs.webcomcpq.com/fed/test/Login.aspx?quote=CAC7C7D2C7

**PDA approval notification tag: `<*PDAAPPROVAL*>`**
- Classic environment result link: https://www.rs.webcomcpq.com/Login.aspx?pdaapproval=C8CFC9D2A89A9D9C9092AEBED2CAC7C7D2C8
- Federation environment result link: https://www.rs.webcomcpq.com/fed/test/Login.aspx?pdaapproval=C8CFC9D2A89A9D9C9092AEBED2CAC7C7D2C7

**Related Information**

Federation Troubleshooting
Federated Single Sign-On (FSSO) [page 204]
Federation Settings [page 206]

### 3.5 Federate Multiple Tenants with a Single AD FS IDP

Follow the provided steps to set up multiple tenants on AD FS (Active Directory Federation Services) IDP.

**Procedure**

1. Log into SAP CPQ Tenant A.
2. Go to Setup Providers Federation Settings.
3. Select the identity provider from the Select Identity Provider dropdown.
4. Scroll down to the Routings panel press Edit.
5. Press Add to add a new row.
6. In Sign on relay state column, select the currently used tenant.
   The name of the URL is automatically generated.
7. Press Save at the bottom-right corner of the screen (for the entire Federation Settings page).
   The new route is generated.
8. Press *Activate* at the bottom of the page.

**Next Steps**

Repeat these steps for the other tenants (Tenant B).
4 AdobeSign Integration

AdobeSign integration enables users to easily sign electronic documents and contributes to a faster closing of deals by sales representatives.

The integration is enabled by administrators for the respective users. For the integration to be successful, the email address of an SAP CPQ user needs to match the one in AdobeSign. Otherwise, an error message displays.

Prerequisites

The following prerequisites need to be completed for this integration:

- an AdobeSign account is created at AdobeSign portal.
- Uncertified applications are enabled to access your AdobeSign account (applicable only for demo/development/sandbox environments). To enable the access, log into AdobeSign portal and access Account. Under Account Settings, open Security Settings. Select the following options under API: Allow Uncertified partner applications to access data from this account and Allow users who aren’t account administrators to develop applications that use the Adobe Sign API.

Complete the steps in Set Up Integration on SAP CPQ Side [page 219] to set up the integration.

Document Template Tags

AdobeSign tags are used to indicate locations in the document where recipient needs to act. If tags are added to the document, then signer is forced to sign the document at the exact place in the document where the tag has been placed. If tags aren’t added to the document, then AdobeSign defines the place where the signer places their signature.

E-Signing Templates

To mark place in the document generation template where signer should put their signature, you must insert AdobeSign tag as a signature placeholder: {{Sig_es:signer1:signature}}.

Integration Statuses

There are four quote statuses that are used during the signing process:

- E-Sign Accepted
- E-Sign Declined
- E-Sign Rejected
Waiting for E-Signing

When the quote is sent to the representative for e-signing, the quote status is changed to Waiting for E-Signing. As soon as the document is signed, the quote status moves to E-Sign Accepted. On the other hand, if the signer refuses to sign the document, it’s moved to E-sign declined. If the document is revoked before it has been signed, the quote status changes to E-Sign Rejected.

Integration Actions

Transition between e-sign statuses is performed via the integration actions. For this purpose, four built-in actions are available:

- **Generate Document for E-Sign**
  
  This action is the only one from this group that is visible in quote layout and that can be triggered by the SAP CPQ user. This action isn’t supposed to change the quote status. In order to make this action visible in the quote layout, you must enable it by setting the workflow permission for it in Setup > Workflow/Approval > Workflow Permissions.

- **Change E-Sign Status**
  
  This action is automatically triggered after generating the document, when the user sends a document to AdobeSign. If the document processing is successful, the action is triggered and the quote moves to the following status.

- **E-Sign Accepted**

- **E-Sign Declined**
  
  The actions E-Sign Accepted and E-Sign Declined are triggered by AdobeSign and aren’t visible in the quote layout. Based on user interaction through AdobeSign, one of the two actions is triggered. The workflow permissions must be set for these actions in Setup > Workflow/Approval > Workflow Permissions. Additionally, you must also set the workflow for these actions in Setup > Workflow/Approval > Workflow My Quotes in the following way:

  - The Generate Document for E-Sign action isn’t supposed to change the quote status. After the quote is sent to AdobeSign using the email form that appears on the screen after the document is generated, the quote status is supposed to change from any status to Waiting for E-Signing.

  - The E-Sign Accepted action should change the status from Waiting for E-Signing to the E-Sign Accepted status.

  - The E-Sign Declined action is supposed to change the status from Waiting for E-Signing to the E-Sign Declined status.

Guest User

The AdobeSign process is performed by the SAP CPQ guest user. The guest user is set under Setup > Quotes > Email Customers / Quote Acceptance Settings > User used for Quote Acceptance. The user type that the guest user belongs to must have permissions for the AdobeSign Accepted and AdobeSign Declined actions.
Multisign Process

The SAP CPQ integration with AdobeSign supports signing by more than one signer. This process is called Multisign. This section provides you with the information what needs to be done on the SAP CPQ side to enable the multisign process.

The only setup activity needed to enable the multisign process is to add signature placeholders for all signers to the document generation template. If this step is omitted, the multisign process is still available and functional. The reason for that is the fact that AdobeSign automatically locates the placeholders at the bottom of the document template in case when no signature placeholders were found. However, the general recommendation is to insert placeholders into the template documents at the desired place in order to keep the document generation templates well-organized and properly formatted.

No workflow changes are supposed to occur since the quote moves to status “Waiting for E-Sign” upon being sent to signers and remains in that status until all signers sign or decline to sign the document. The placeholder that is used to mark where signers should sign is: {{Sig_es_:signer{r}:signature}}, where {r} should be the ordinal number of the signer (for example {{Sig_es_:signer3:signature}}). A detailed example is provided in AdobeSign Signing Process Example [page 218].

AdobeSign Document Revoke

SAP CPQ supports the Revoke Document feature, for users who want to void a document that has already been sent to signers, avoiding the need for signers to review an obsolete document. This action can be triggered from the quote since it has been developed as the standard quote action Void Document. Workflow should be modeled to move the quote back from Waiting for E-Signing to the previous status, after the Void Document action is triggered. Once the signer clicks on the link received via email, they are informed that the document has been revoked.

Add CC

To send a copy of the signed document to additional recipients for informational purposes, click Add CC and enter the email address of the additional recipient in the designated field. If there are multiple recipients in the Add CC field, separate their email addresses with a comma. The recipients entered into the Add CC field receive two emails:

- The first email is received when the document is sent for signing.
- The second email is received after all signers sign the document. This email contains a copy of the signed document.

Admin Mode for Sending Documents to Adobe Sign

The Admin mode enables you to send documents to Adobe Sign as the administrator, instead of having the currently logged-in user act on behalf of the administrator. To set up the integration in the administrator mode, follow the procedure:
1. Go to [Setup ➤ Providers ➤ Adobe Sign ➤ ]
2. Set Use Admin Mode to TRUE. The Impersonate Email field displays.
3. Enter a new email address that is visible to the signing party.
4. Click Save.

Every time a document is sent to Adobe Sign, regardless of the logged-in user, the document originates from the defined email address. However, if no email address has been set up, the document originates from the email address SAP CPQ administrator used to create the SAP CPQ account.

Example

Company B, a partner of company A, has access to SAP CPQ and needs to send SAP CPQ documents to Adobe Sign for signing. In this case, SAP CPQ sends documents to company A's Adobe Sign account on behalf of company B. If company B already has an account at Adobe Sign, company A won’t be able to send SAP CPQ documents to Adobe Sign because the same email address was used when acting on behalf of company B and when company B opened their own account. However, once company A sets the administrator mode for the Adobe Sign integration, they will be able to send SAP CPQ documents to Adobe Sign for signing on behalf of company B.

AdobeSign Signing Process Example [page 218]
By following this procedure, you can test the signing process with the AdobeSign integration.

Set Up Integration on SAP CPQ Side [page 219]
The integration with AdobeSign needs to be set up on the SAP CPQ side.

4.1 AdobeSign Signing Process Example

By following this procedure, you can test the signing process with the AdobeSign integration.

Procedure

1. Create a SAP CPQ quote and click Generate Document for E-Sign.
2. Choose a document generation template and click on it to generate the document.
   Once the document is generated, the email form is displayed. The email form is used to send the document to the signers for signing.
3. Define the signers by clicking Add Signer and entering the signer’s email, name, position, and company name.
   Only email and name fields are mandatory. The Add Signer button allows you to select Bill to, Ship to, or End user role as the signer, provided that you have the role defined on the quote.
   You’re able to define more than one signer. If you define at least two signers, you’re allowed to choose if they sign the document in a defined order or sequence, or not. If you select this option, the document link
is sent to only one signer - the first in the list. As soon as they sign, the document link is sent to the next signer. If the first signer rejects to sign the document, the signing process is terminated, so the following signers don’t receive the link.

4. To send the document to signers, click Send agreement to AdobeSign.

The signer receives an email with the link to the document. Clicking the link leads signer to the AdobeSign service page where they can sign. If your document contained signer placeholders, then the signer is positioned at the appropriate placeholder to sign the document.

5. Click on the signing box (Click here to sign).

A pop-up window opens, where the signer can choose whether they want to type, draw or upload an image that contains a scanned signature.

6. Click Apply.

The pop-up window is closed.

7. Click Click to Sign.

After signing the document, the signer is offered to download a copy of the signed document.

Once the document is signed, you can see who signed it and when. To view who signed the document, return to quote and go to the Documents tab to locate the generated quote document. The Additional Info column displays the following information:

- the date and time the document was sent for signing
- the names and email addresses of the designated signers
- the date and time of signing
- the provider used to sign the document (AdobeSign)
- the names and email addresses of the users who signed the document

If a signature request is declined, the column displays the date and time the request was declined, the name and email of the person who declined the request, and the reason for declining to sign.

**Note**

In order for the Additional Info column to be visible, the Additional Info Visibility application parameter must be set to TRUE.

### 4.2 Set Up Integration on SAP CPQ Side

The integration with AdobeSign needs to be set up on the SAP CPQ side.

**Procedure**

1. Go to Setup > Providers > Providers > AdobeSign >.

AdobeSign Provider Settings opens.

2. Click Set Admin Account to set the integration parameters for your tenant.
A pop-up window opens.

3. Enter your email and password in the pop-up.

4. Click Sign In.

   An informational pop-up displays stating the access rights are enabled by the integration.

5. Click Allow Access.

   A message informs you that you have successfully logged into AdobeSign.

6. Click Close to proceed further.

   You are now redirected to the settings page.

7. Select TRUE in the Integration turned on dropdown menu.

8. Click Save.

   The integration setup is now complete.
The integration between SAP CPQ and SAP CLM allows sales representatives to generate contracts in SAP CPQ and send them to SAP CLM for negotiation and redlining.

CPQC integration is developed to ensure effective communication between sales representatives (SAP CPQ) and legal teams (SAP CLM).

SAP CLM is a quick, user-friendly solution that automates contract management from inception to completion. SAP CLM offers many robust features to increase collaboration and accelerate deal cycles. For more details on SAP CLM, visit the SAP CLM Help portal.

The communication between SAP CPQ and SAP CLM is bidirectional. After sales representatives generate a contract with quote items and send it to SAP CLM, legal teams can review and redline the contract. They leave comments on the contract in case changes are needed and send the comments to SAP CPQ. On their side, sales representatives can either update the contract according to comments, or respond to comments and in that way prolong the final agreement. While the negotiation is ongoing and changes are being made to quote items, users on both sides work only on one contract - the one that was first generated.

**Setup Prerequisites** [page 221]
Before sales representatives and legal teams can exchange quotes and contracts in the CPQC integration, administrators must adjust the respective settings on both sides.

**CPQC User Side** [page 225]
This topic describes the integration with SAP CLM on the user side.

**Typical Business Scenario** [page 226]
In CPQC, you can map statuses and set up actions according to your specific business model. The following example describes a typical use case when the initially generated contract needs to be adjusted and the negotiation goes back and forth between SAP CPQ and SAP CLM until a final agreement is reached.

## 5.1 Setup Prerequisites

Before sales representatives and legal teams can exchange quotes and contracts in the CPQC integration, administrators must adjust the respective settings on both sides.

### SAP CLM Integration User

An integration user must be created in SAP CLM to enable the user to seamlessly move between products and perform the integration tasks. The integration user should at least have the permission to view quotes. Additional actions (such as quote editing) can be enabled for the user as well, depending on the SAP CPQ model. Users are matched by the email or username (the username of the SAP CLM user must match the...
email of the SAP CPQ user). Contact SAP CLM support team to have them create an integration user for your CPQC. You will receive integration details that you must enter in the provider settings in SAP CPQ.

**Actions and Workflow [page 222]**

Actions need to be added to the appropriate workflow matrix and permissions for these actions need to be set.

**Provider Settings [page 223]**

The main adjustments for enabling the integration in SAP CPQ are managed in the Providers section.

### 5.1.1 Actions and Workflow

Actions need to be added to the appropriate workflow matrix and permissions for these actions need to be set.

- **Send Quote To CLM** - the action button is displayed on the right side of the email form for sending the generated document. The action opens the document in SAP CLM in a new tab. The action should be available only after the quote reaches the status that is the result of the Generate Document action. Set it in the matrix after the Generate Document action (you first want to generate a document from a quote and then send it to SAP CLM).

- **Update Quote in CLM** - the action sends all the changes made on a quote in SAP CPQ to the respective contract in SAP CLM. You should send updates to SAP CLM after a document has been generated and sent to SAP CLM. There must be a contract in SAP CLM originating from the quote for the system to properly apply changes.

- **Add Comment from CLM** - the action isn’t visible in the SAP CPQ’s user interface. However, you should add it to the workflow matrix to allow retrieving comments from SAP CLM. The action needs to be enabled in the status in which the quote receives comments.

- **Change Status In CLM** - the action changes the status of the contract in SAP CLM, as defined in Status Mappings. Add the action to the workflow matrix according to your model of SAP CPQ and set the workflow permissions.

- Change Status From CLM - the action isn’t visible in SAP CPQ’s user interface. Changes the status of the contract in SAP CLM, as defined in Status Mappings. Add the action in the workflow matrix according to your model of SAP CPQ and set the workflow permissions.

### Related Information

- Setup Prerequisites [page 221]
- CPQC Integration [page 221]
5.1.2 Provider Settings

The main adjustments for enabling the integration in SAP CPQ are managed in the Providers section.

Prerequisite

- to have an integration user created in SAP CLM.

Procedure

1. Go to Setup > Providers > Providers. A page with available providers displays.
2. Click the SAP CLM card. A page with two tabs displays. Provider Settings is open by default.
3. Fill out the fields with data you received upon SAP CLM tenant creation.
   - Use CLM Integration - set to TRUE to enable the integration.
   - CLM Integration URL - URL of the SAP CLM environment where your tenant was created.
   - CLM integration username - email of the SAP CLM integration user. The integration authorization is based on this value.
   - CLM integration password - password of the SAP CLM integration user.
4. Click Save. The SAP CPQC integration is now enabled.

Template Mappings

Prerequisites

- to have a contract type created in SAP CLM.
- to have an active document generation template created in SAP CPQ.
- to have these two tags in the SAP CPQ template: $QUOTE_LINE_ITEMS and $OPTIONAL_QUOTE_LINE_ITEMS. The tags retrieve line and optional quote items that are displayed in the contract. Alternatively, you can add the tags directly in the contract in SAP CLM. You should insert the tags only once per document for the system to process them properly. These aren't the standard SAP CPQ tags for retrieving quote items in the document.

Procedure

1. On the CLM provider settings page, click Template Mappings. The page lets you map an SAP CPQ template to an SAP CLM contract type (for example, Master Service Agreement (MSA)).
2. Click Add New. Enter the name and the description of the mapping.
3. Click next to the dedicated fields to search for a document template and a contract type, respectively.
4. Click Save. The SAP CPQ template is mapped to the SAP CLM contract. New tabs display to map quote item columns and statuses for the respective template.
Field Mapping

The Field Mappings SAP CPQ to CLM and Field Mappings CLM to SAP CPQ tabs are available only when you are within a template or contract mapping. In the Field Mappings SAP CPQ to CLM tab, map SAP CPQ fields to SAP CLM fields (the value of an SAP CPQ field displays in the mapped SAP CLM field in the contract). SAP CPQ fields are grouped under entities (Quote, Visitor, and so on). When adding a new field, you first need to select an SAP CPQ entity and then a related field. Both standard and custom quote fields are available. In CLM Field, first select one of the SAP CLM objects (Contract, Customer, and Contact) and then one of the related standard or custom fields.

**Note**

In the list of SAP CPQ entities, choose Tag (Dynamic Formula) if you wish to dynamically retrieve data from the system. Create a formula in the standard SAP CPQ formula builder.

Although you can map any fields you wish the legal team and customers to see in the contract, some fields are required on the SAP CLM side: name (Customer), email (Contact), contactFirstName (Contact), and contactLastName (Contact). These fields can be mapped to any fields in SAP CPQ (for example, Bill to Customer).

In Field Mappings CLM to SAP CPQ, choose an SAP CLM object and a field that will be mapped to an SAP CPQ field. The values of SAP CLM fields can be mapped only to quote custom fields in SAP CPQ.

Quote Item Column Mappings

The tags in the document retrieve values from quote item columns selected in the Quote Item Column Mappings tab. Move the columns from available to selected and vice versa by clicking on them. The following columns are selected by default and you can’t remove them from the list: Description, CartItem and RolledUpCartItem.

Some product columns aren’t available in the list and can’t be sent from SAP CPQ to SAP CLM:

- Upfront Percent
- Upfront Amount
- Overall Discount Percent
- Overall Discount Amount
- Total Shipping Cost
- Item Delivery Method
- Item Delivery Status
- Monthly Fee Amount
- Sub Total Amount
- Total Weight
- MPN
- Family Code
- Lead Time
- Inventory
- ProductVersion
Status Mappings

In Status Mappings, administrators define the lifecycle of a quote and the contract in the two systems. In this tab, you define how statuses will change from SAP CPQ to SAP CLM and vice versa after an action is executed. It’s possible to map a status in one application to a status in the other application. The statuses change when the Change status in CLM and Change status from CLM actions get executed. When the agreement on quote items is reached, SAP CPQ users change the status of the quote to a status that indicates that the negotiation is completed. Additionally, SAP CPQ users need to execute the Change Status in CLM action so that the status of the contract in SAP CLM would change accordingly. SAP CLM Side To configure provider settings in SAP CLM, find detailed instructions on the SAP CLM Help Portal.

Related Information

Setup Prerequisites [page 221]
CPQC Integration [page 221]

5.2 CPQC User Side

This topic describes the integration with SAP CLM on the user side.

Listed here are some facts about the quote behavior in the integration environment:

- All product types are retrieved via the $QUOTE_LINE_ITEMS and $OPTIONAL_QUOTE_LINE_ITEMS tags and displayed in the contract.
- Quote revision contains the comments made on the source quote up until the revision was created.
- The Copy action copies only the content of the quote, not the comments. The same applies when copying items.
- When a quote is reassigned, the comments are visible to the new quote owner.
- For autocomplete attributes, the name of the value code column is sent to SAP CLM.
- For drilldown attributes, the value from the Setup, not from the configurator is sent to SAP CLM.
i Note

To enable users to comment on quotes that have been assigned to them, add the Add Comments from CLM action to the workflow matrix in the Other Quotes tab. After a quote is reassigned, the new assignee needs to trigger the Update Quote in CLM action to send new item IDs to SAP CLM.

Comments

In the integrated environment, the Collaboration/Comments button displays on quotes for which a document is generated and sent to SAP CLM. The button displays the number of unread comments that are sent from SAP CLM. Clicking the button opens the Comments chat window where you can reply to the comments sent from SAP CLM.

The system lets you create a general comment when commenting on the quote in global. In addition, you can also create specific comments for each item. The maximum number of characters in one comment is 2000 in any alphabet. Saving the quote updates the comments.

Related Information

Setup Prerequisites [page 221]
Typical Business Scenario [page 226]

5.3 Typical Business Scenario

In CPQC, you can map statuses and set up actions according to your specific business model. The following example describes a typical use case when the initially generated contract needs to be adjusted and the negotiation goes back and forth between SAP CPQ and SAP CLM until a final agreement is reached.

After a quote with items is created in SAP CPQ, the sales representative generates a document using the template that is mapped with a contract template in SAP CLM. Depending on the business process, the quote
in SAP CPQ may go through different statuses. After the document is sent to SAP CLM, the user executes the action Change Status in CLM to align the lifecycle of the quote and the generated contract.

The legal team reviews the contract, redlines it and adds suggestions in the comments section in Commercial Details. The team sends the contract to customers who add their comments in case any change of the contract is needed (change of price, higher discount, number of items, and so on).

While the contract goes through review in SAP CLM and changes statuses accordingly, the status of the quote in SAP CPQ doesn’t change. When the contract reaches the status that was mapped in Status Mappings in SAP CPQ, the action Change Status from CLM is triggered and the status of the quote changes accordingly. The user in SAP CPQ gets a notification that the status of the contract has changed.

The Collaboration/Comments button in SAP CPQ shows the number of comments that are sent from SAP CLM. The sales representative makes the changes according to SAP CLM comments, replies to comments and saves the quote. The replies are automatically sent to SAP CLM and users can view them in the All Comments section in Commercial Details.

When both parties agree on the offer and the contract terms, the status of the contract in SAP CLM is changed to the target status and, consequently, the quote in SAP CPQ will reach the right status.

**Related Information**

CPQC User Side [page 225]

Setup Prerequisites [page 221]
6 DocuSign Integration

DocuSign integration enables you to send a quote to customer, so they can e-sign the document and send back to sales representative. This shortens the time needed for the entire process to be completed and allows sales representatives to close their deals faster.

The instructions below explain how to integrate a demo tenant with DocuSign. Please note that DocuSign recommends that you do not add files larger than 25MB to an envelope.

**Note**

If SAP CPQ has already been integrated with DocuSign prior to may 2017 Release, some changes within DocuSign should be performed. Prior to this Release, when integrating SAP CPQ and DocuSign, a DocuSign custom field had to be created through the DocuSign setup process. That custom field had the same format as placeholder \signhere_{r}\. This custom field is not needed anymore since all the mappings are performed automatically. Therefore, customers with an existing SAP CPQ - DocuSign integration need to delete this custom field within DocuSign. If it remains in DocuSign, the first signer of each document will have to sign the document twice.

6.1 Example and Use Cases for Signing Process

To test DocuSign process, create new SAP CPQ quote and click on the Generate Document for E-Sign action. The document sending form will be displayed as the result.
Clicking the **Send to DocuSign** button, email with link to document for signing is sent. Make sure that you send this email only to one recipient, because the link can be used only by one recipient during DocuSign process. As soon as document is signed (DocuSign process completed), link expires and becomes invalid. Success message is displayed at the top of the form. Quote status changes to *Waiting for DocuSigning*.

![Email Form](image)

Click the link to access DocuSign. You are redirected to the DocuSign page where you should agree to use electronic records and signatures and decide which action to undertake. There are a few actions available:

- Sign the document
- Decline to sign the document

---

Hi Sandra, I am pleased to present the attached response to your search. Our extensive experience in solving business issues such as the ones presented in our discussions allows us to have a very high degree of confidence in our ability to solve your issues. I welcome any inquiries about our solutions, their capabilities, or our ability to integrate it into your current or planned business process. Please click the link below to download your proposal. Best Regards, John Davis

Please sign document with DocuSign by clicking [here](https://dev.webcomcpq.com/DocuSign/DocuSignInitiate.aspx?dscid=Nzg3MGQ1ZWItYmRkMy00NmFiLWFiNTUtNzdmM2RkYThkMTVj&dsd=V0VCQ09NUUE=).

If you are not able to follow the above link, copy and paste the following text into your web browser:

```html
https://dev.webcomcpq.com/DocuSign/DocuSignInitiate.aspx?dscid=Nzg3MGQ1ZWItYmRkMy00NmFiLWFiNTUtNzdmM2RkYThkMTVj&dsd=V0VCQ09NUUE=&w
```

Click the link to access DocuSign. You are redirected to the DocuSign page where you should agree to use electronic records and signatures and decide which action to undertake. There are a few actions available:
• Sign on paper and upload or fax the document back
• Finish signing process later

Use case 1 - You choose to accept to sign the document

If you did not put signature placeholders in your document earlier, you can choose a place in the document where to sign. Click Signature on the left side and drag and position your signature. You can also choose to put your initials or name instead. There are a few more options that can be added (date, email address, company name, and so on).

When you are done, click the Finish button. This way, the process is complete.

The generated and signed document appears in the Documents tab on the quote. Quote status is updated to DocuSign Accepted.

The Additional Info column in the Documents tab displays the following information:
• The date and time the document was sent for signing
• The name and email address of the designated signer
• The date and time of signing
• The provider used to sign the document (DocuSign)
• The name and email address of the user who signed the document

**Note**

The visibility of the Additional Info column depends on the Additional Info Visibility application parameter, which has to be set to TRUE. The parameter can be accessed in Setup > General > Application Parameters > Shopping Cart and Quotes.
Use case 2 - You choose to decline to sign the document

If you don’t want to sign the document, select the appropriate action from the action list.

Clicking on the **Decline to sign** action, you are prompted to choose whether you want to decline the document immediately or finish later. If you click the **Continue** button, a new prompt is displayed where you can enter your comment (comment is not required).

Clicking the **Decline to sign** button, the quote status changes to the **DocuSign Declined** status. In addition, the **Additional Info** column displays the date and time the request was declined, the name and email of the person who declined the request as well as the reason for declining to sign the document.

Use case 3 - You choose to sign the document manually

If you prefer to sign the document manually, click the **Print & Sign** button.
Once you click the button, a new prompt displays, where you should choose how you are going to return the document - whether by uploading or sending the document by fax. Moreover, you can also download, and then upload the document back.

### 6.2 DocuSign Side

To get started, you must create DocuSign Developers account. Go to [DocuSign Developer Link](#) for instructions on how to create an account. You’ll receive an email from DocuSign to activate the account.

DocuSign integration account/user needs to have two settings checked on the account otherwise the integration WILL NOT work. This has to be done by DocuSign Support so a support ticket has to be logged to DocuSign if these settings aren’t checked for the integration user.

- In-session- Checked (that is, Enabled)
- Certificate Enforcement Status - Checked (that is, Don’t Enforce In Session Certificate)

### DocuSign Envelope Expiration

When document is sent from SAP CPQ to DocuSign, DocuSign Envelope is created and its expiration time is set. The default expiration period is 120 days. This can be changed in [DocuSign Settings > Preferences > Account Administration > Reminders & Expirations](#).

For more details, refer to this article: [DocuSign - Reminders and expirations](#).
DocuSign Time Out

Clicking on link received through email, DocuSign recipient token is created for current recipient / signer. Recipient token expires five minutes after it’s issued by DocuSign. If a recipient token URL is accessed after token expiration, the user is redirected to SAP CPQ callback URL with event code “TTLExpired”.

For more details, please refer to this page: DocuSign Time Out

Set up API Request Logging

To follow all requests and responses to and from DocuSign API, you should enable API logging. To do so, on in DocuSign Settings go to Preferences > Member Options > Permissions. Under the DocuSign API Options group, select the Enable API Request Logging checkbox.

6.3 SAP CPQ Side

1. Go to Setup > Providers > Providers. A page containing available providers appears.
2. Click DocuSign. A page where you can set up DocuSign for your tenant displays.

i Note

Integration parameters - email, password, and login URL must be provided by DocuSign support. Other than these three parameters, you also need the Integrator key to be set for your tenant. This is performed by SAP CPQ support. When requesting the integrator key let the support know whether your tenant is on sandbox or production environment.

As soon as you are provided with the information needed, proceed by entering email, password, and login URL into the appropriate fields and click Save.
iNote

The email accompanying generated documents that are going through the DocuSign e-signature process no longer uses the email template defined in Setup > Quotes > Email Customers/Quote Acceptance Settings. Subject and body of emails accompanying generated documents signed with DocuSign are now defined in Setup > Providers > Providers > Docusign > General Settings.

Document Template Tags

DocuSign® tags are used to indicate locations in the document where recipient needs to take action. Also, some tags are used to provide recipient with some information. Some tags are automatically filled with recipient’s information.

If tags are added to the document, then signer is forced to sign the document at the exact placeholders in the document (where predefined tag is placed). If tags are not added to the document, then signer/recipient is free to choose place where they will put their signature. In this case, DocuSign will offer a list of all available tags (for example, Initials, Signature, Date, and so on), so signer can decide what to put where.

Creating Templates for DocuSigning

To mark place in the document generation template where signer should put their signature, you must insert placeholder \signhere_1\.
Since placeholders will not be replaced by DocuSign (the signatures will be put over them), the text color of placeholders should be set to white (or any other background color that the document has), which will make them invisible in the template document and in the signed document.

Integration statuses

Integrating SAP CPQ and DocuSign, three new statuses must be defined by SAP CPQ administrators. These statuses will be used in DocuSign process, as starting and ending statuses of DocuSign integration actions. Three statuses can be named as follows, but SAP CPQ administrators are free to pick their own status names:

- DocuSign Accepted
- DocuSign Rejected
- Waiting for DocuSigning

Integration actions

DocuSign integration brings four built-in actions that are available in SAP CPQ Setup:

- Generate Document E-Sign
- Change E-Sign Status
- DocuSign Accepted
- DocuSign Rejected

Action "Generate Document E-Sign"

The only action from this group that is visible in quote layout and that can be triggered by SAP CPQ user is Generate Document For E-Sign. Note that this action should not change the quote status. In order to make this action visible in quote layout, you must enable it by setting a Workflow Permission for it.

To do so, go to Setup ➤ Workflow/Approval ➤ Workflow Permissions and assign permission for this action to the particular user type.

Action "Change E-Sign Status"

This action is automatically triggered after generating the document - in the moment when user sends a document to DocuSign. If document processing is successful, the action is triggered and quote moved to the next status. Otherwise, action is not triggered and quote status is not changed.
Actions "DocuSign Accepted" and "DocuSign Declined"

These actions are triggered by DocuSign and also not visible in quote layout. Based on user interaction through DocuSign, one or another action will be triggered. You should set permission for these two actions too, as described earlier. Also you must set workflow for these actions.

To do this, go to Setup Workflow/Approval Workflow, and under tab My Quotes set actions workflow as described under the section that follows.

Set Up Workflow to Support the DocuSign Process

- Action Generate Document For E-Sign itself should not change the quote status. After the quote is sent to DocuSign using the email form that appears on the screen after the document is generated, quote status should change from Default, Open or any other status to Waiting for DocuSigning status.
- Action DocuSign Accepted should lead quote from Waiting For DocuSigning to DocuSign Accepted status.
- Action DocuSign Declined should lead quote from Waiting For DocuSigning to DocuSign Declined status.

Guest User

DocuSign process is performed by an SAP CPQ guest user. Guest user is set in Setup Quotes Email Customers/Quote Acceptance Settings.

Guest user should belong to the user type that has permissions for the DocuSign Accepted and DocuSign Declined actions.

6.4 Multisign Process

The SAP CPQ integration with DocuSign supports signing by more than one signer. This is called multisign process. The following section provides you with the information what should be done on SAP CPQ side to enable the multisign process.

The only setup activity needed to enable the multisign process is to add signature placeholders for all signers to the document generation template(s). Note that if this step is omitted, the multisign process will still be available and functional. That is because DocuSign enables users to put their signature anywhere in the document if no signature placeholder was found. However the general recommendation is to insert placeholders into template documents at the desired place in order to keep the document generation templates well organized and formatted.

No workflow changes should be performed since the quote moves to status Waiting for DocuSign upon being sent to signers and remains in that status until all signers sign or decline to sign the document. The placeholder that is used to mark where signers should sign is: \signhere_{r} where \{r\} should be the ordinal number of the signer. The detailed example is provided in the Examples section of this page.
Besides `{signhere}` tag, there are three more tags that can be placed into the document generation template which will provide more information about the signers and countersingers.

- `{signer_{r}_name}` - Inserts name of the signer
- `{signer_{r}_company}` - Inserts company of the signer
- `{signer_{r}_position}` - Inserts position of the signer

**Multisign Process Example**

In the example, two placeholders are added to the document generation template where two signers should sign the document.

Since placeholders will not be replaced by DocuSign (the signatures will be put over them, the text color of placeholders was set to white, which made them invisible in the template document and in the signed document.)
THE COMPLETE AGREEMENT BETWEEN THE COMPANY AND BUYER IS CONTAINED HEREBIN AND NO ADDITIONAL OR DIFFERENT TERM OR CONDITION STATED BY BUYER SHALL BE BINDING UNLESS AGREED TO BY THE COMPANY IN WRITING.

No course of prior dealings and no usage of the trade shall be relevant to supplement or explain any terms used in this Agreement. This Agreement may be modified only by a writing signed by both the Company and Buyer and shall be governed by the Uniform Commercial Code as enacted by the State of Wisconsin. The failure of the Company to insist upon strict performance of any of the terms and conditions stated herein shall not be considered a continuing waiver of any term or condition or any of the Company’s rights.

Clicking on the Generate Document for E-sign action and picking the particular Document Generation Template, an email form for sending the document to signers gets displayed.

The form is used to define both the signers and countersigners.
The multisign feature also allows users to instruct SAP CPQ to request signers to sign in a defined order. If the Complete in Order toggle switch is enabled, an email with the DocuSign link is sent to signers successively, not at once. This means that the signer will receive the link only after all signers listed before them sign the document. Alternatively, if the toggle switch is not enabled, the DocuSign link will be sent to all signers at once.

It is also possible to mark a signer as required. The image below shows that two signers are defined and that the Complete in Order toggle switch is active. One signer is required, whereas the other is optional.

When you click Send to DocuSign, the designated signer gets the email with the link to DocuSign. Upon clicking the link, the user will be redirected to DocuSign. Clicking Next, DocuSign will position the cursor at the placeholder that was previously defined in the document template.

Once positioned, the user can sign the document by clicking the Sign button placed over the placeholder.
Users can then select whether to put their full name or initials into the signature.

Clicking *Adopt and Sign* places the signature over the placeholder. The last step is to click *Finish*. After the first signer has signed the document, an email will be sent to the next signer. Even though the second signer is not marked as required, the signer has to click the link received via email and then click *Finish* on the DocuSign side, whether he signed the document or not.
Once the second optional signer clicks the **Finish** button, the document obtains the **DocuSign Accepted** status, and the DocuSign process is completed.

### 6.5 Countersign process

The countersign process is used by companies that want to use SAP CPQ to also put their own signatures on the document. To set up the process, signers must be defined through SAP CPQ administration first. In SAP CPQ terminology these signers are known as “countersigners”.

As an administrator you define a list of potential countersigners and then after generating the document users select which (if any) countersigners are needed for the particular document.

Countersigners are defined in [Setup > Providers > Providers > DocuSign](#).

Other than the list of countersigners, the SAP CPQ administrator should modify the **Document Generation Template** by adding placeholders where countersigners will sign the document. The placeholder format is `\countersign_{r}\`, where `{r}` should be the ordinal number of the countersigner. Countersign placeholders, definitions, and signing processes are the same as described in the multisign section above.

#### Countersign process example

In the example, two placeholders are added to the document generation template where two countersigners should sign the document.
Since placeholders will not be replaced by DocuSign (the signatures will be put over them, the text color of placeholders was set to white, which made them invisible in the template document and in the signed document.

Clicking on the **Generate Document for E-sign** action and picking the particular **Document Generation Template**, a new email form for sending document to signers and countersigners appears.

The SAP CPQ user has the ability to control if the countersigners should sign the document before signers and if countersigners should sign the document in the defined order.

In addition, it is also possible to mark a countersigner as required. As shown in the image below, two countersigners are defined, and the **Complete in Order** toggle switch is not enabled. One countersigner is required, whereas the other one is optional.
When you click the **Send to DocuSign** button, an email with DocuSign link is sent to countersigners first. Since the **Complete in Order** toggle switch is not active, both countersigners will receive the email at the same time.

The required countersigner will have to sign (or decline) the document. The optional countersigner does not need to sign the document. However, the optional countersigner will have to open the received link and click the **Finish** button, so the process can continue. The image below shows the situation when the required signer has already signed the document and the optional countersigner clicks the button to complete the signing process for countersigners.

After countersigners complete the signing process, the link is sent to signers. The particular SAP CPQ quote is still in the **Waiting for DocuSign** status since the signing process has not yet been completed on the signers’ side.

The process is completed once the required signer signs the document and the optional signer either signs or clicks **Finish**. The quote moves to the **DocuSign Accepted** status.

### 6.6 DocuSign Document Revoke

SAP CPQ supports the **Revoke Document** feature, for users who want to void a document that has already been sent to signers, avoiding the need for signers to review an obsolete document. This action can be triggered from the **Quote** page since it has been developed as standard SAP CPQ quote action. Action name is **Void Document**.

Workflow should be modeled to move the quote back from **Waiting for DocuSigning** to the previous status, after the **Void Document** action is triggered. Once the signer clicks on the link received through an email, they will be informed that the document has been revoked.

When the SAP CPQ user generates a new document, a new link will be sent to the signer.
7 MS Dynamics Integration

SAP CPQ can be integrated with the MS Dynamics CRM solution.

**Note**
All mapping of entities in the two systems should be handled via IronPython scripting, and not in the SAP CPQ Setup.

**MS Dynamics CRM and SAP CPQ Integration**
This topic provides you with instructions how to successfully integrate your MS Dynamics CRM with SAP CPQ.

**MS Dynamics CRM (2011 version) and SAP CPQ Integration Instructions**
SAP CPQ can be integrated with MS Dynamics CRM (2011 version).

**Dynamics CRM Integration Instructions for MS CRM 4.0**
In order to access Cpq Quotes through MS CRM opportunity, you need to log in to MS CRM with administrative account.

7.1 MS Dynamics CRM and SAP CPQ Integration

This topic provides you with instructions how to successfully integrate your MS Dynamics CRM with SAP CPQ.

**Prerequisites**

Federated SSO needs to be set up for the integration to work properly. The following steps need to be taken in order to establish SSO between SAP CPQ and Microsoft Dynamics CRM:

- SAML metadata files exchange between IdP and SP side. ADFS or Azure AD is the identity provider (IdP) on the customer side and SAP CPQ is the service provider (SP).
- User mapping should be configured on ADFS side and SAP CPQ side.
  When SSO is configured, a CRM user will automatically be mapped to a SAP CPQ user after landing from CRM to SAP CPQ and there won’t be any need to send the ID of the CRM user in the URL of the SAP CPQ landing page.

**Note**
All mapping of entities in the two systems should be handled via IronPython scripting, and not in the SAP CPQ Setup.
CRM Setup

To start administering MS CRM, you need to have an administrative account. Log in using administrative credentials and then go to Settings and click Customizations. On the next screen, click Customize the System. A list of components is displayed. The first step is to create a new entity called \textit{CPQ\_Quote}.

**Step 1: Create CPQ\_Quote entity**

Among components from the left pane, select \textit{Entities} and click \textit{New}. This brings screen with the form to define \textit{Entity}. Fill the form as instructed:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Field value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name</td>
<td>\textit{CPQ Quote}</td>
<td></td>
</tr>
<tr>
<td>Plural Name</td>
<td>\textit{CPQ Quotes}</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>\textit{new_cpqquote}</td>
<td>This field is auto-populated</td>
</tr>
<tr>
<td>Areas that display this entity</td>
<td>\textit{Sales}</td>
<td></td>
</tr>
</tbody>
</table>

Click \textit{Save} and navigate to \textit{Fields} item under the CPQ Quote entity menu in the left pane. This brings screen where you should add fields that are required for the integration as instructed:

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Data Type</th>
<th>Field Type</th>
<th>Format</th>
<th>Maximum Length</th>
<th>IME Mode</th>
<th>Target Record Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{CPQ Quote}</td>
<td>Primary Key</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\textit{Name}</td>
<td>Single Line of Text</td>
<td>Simple</td>
<td>Text</td>
<td>100</td>
<td>auto</td>
<td></td>
</tr>
<tr>
<td>\textit{Opportunity}</td>
<td>Lookup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Opportunity</td>
</tr>
<tr>
<td>\textit{Link To Quote}</td>
<td>Single Line of Text</td>
<td>Simple</td>
<td>Text</td>
<td>100</td>
<td>auto</td>
<td></td>
</tr>
</tbody>
</table>

**Step 2: Customize Opportunity form**

The next step is to customize the Opportunity form the way it displays the list of related SAP CPQ Quotes. To do so, select \textit{Opportunity} from list of entities and click \textit{Forms} (in the left pane). From the list of active forms that are displayed in the right pane, click form named Opportunity to edit it. Note that you should edit the Opportunity form that has type \textit{Main}, not the form that has type \textit{Quick Create}. Form for adding new tab is displayed. Click on \textit{INSERT} tab at the top menu, then click on \textit{One Column}. New tab is created with its default name \textit{Tab}. Double click on \textit{Tab} label to open \textit{Tab Properties} dialog box. Here you can change label from its default value - \textit{Tab to Opportunity Quotes}. Fill out form fields \textit{Name} and \textit{Label}, and click \textit{OK}. Locate your renamed tab \textit{Opportunity Quotes} and point mouse cursor over the section inside the tab, then click on \textit{Sub-Grid} menu item from main menu at the top of the page. \textit{Sub-Grid} properties window is displayed. Fill out form fields as described:
Step 3: New CPQ Quote

In this section, we will describe the procedure to create a button that will be used to authorize the MS Dynamics user to open SAP CPQ catalog, browse it and add products to the SAP CPQ Quote. This quote will be connected to your MS Dynamics Opportunity you are currently working on.

1. Go to Settings > Customizations > Customize the System.
2. Locate the Opportunity entity at the left pane, then click on Forms submenu, then click on Opportunity (Main) form to edit it.
3. New page for editing the form is displayed. Locate the Opportunity Quotes area and click on to activate it, then click on tab INSERT at the top menu, and finally click Web Resource.
4. Add Web Resource dialog box is open. Click on Search icon and then click on Look up More Records.
5. New dialog box Look Up Record is open. Click New.

Click Text Editor button to open text editor. Click on Source tab to open it. Copy and paste code provided below and then click OK.

```html
<html>
<head>
<title>Create CPQ Quote</title>
<meta>
<script type="text/javascript">
  function NewQuote() {
    var entity = window.parent.Xrm.Page.data.entity;
    var id = entity.getId();
    var name = entity.attributes.get("name").getValue();
    var extrasq = "new_opportunity=" + id + ",new_opportunityname=" + name;
  }
</script>
</head>
<body>
</body>
</html>
```
Web Resource details are displayed in the dialog box.

You are done with creating Web Resource. Click OK. Web Resource appears in Opportunity Quotes section. Save Opportunity form, Publish customizations and refresh Opportunity page.

Step 4: Customize SAP CPQ Quote form

Now, we are going to edit SAP CPQ Quote form in order to display SAP CPQ Catalog. To do so, follow next steps.

1. Go to Settings ➤ Customizations ➤ Customize the System ➤ .
2. Locate the CPQ Quote entity at the left pane, then click on Forms submenu, then click on Information (Main) form to edit it.
3. Click on INSERT tab, then select area in section CPQ Quote, click One Column and finally click IFRAME button.
4. Dialog box for adding new Iframe is displayed. Under General tab fill out the required fields as instructed below.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Field value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>CPQ</td>
<td>iframe_ is prepended automatically</td>
</tr>
<tr>
<td>URL</td>
<td><a href="https://teststable.webcomcpq.com/MsCrm/Login.aspx?opportunityId=value&amp;quoteNumber=value&amp;action=value">https://teststable.webcomcpq.com/MsCrm/Login.aspx?opportunityId=value&amp;quoteNumber=value&amp;action=value</a></td>
<td>opportunityId is the ID of the Opportunity object in CRM. quoteNumber is the composite number of the CPQ quote. Quote composite number can be sent in clear text or masked, using 0xFF XOR mask. action is the quote action. It can take values new, edit or view. new should be used for creating a new quote object. edit should be used for editing an existing quote, and view should be used to access the quote in the read-only mode.</td>
</tr>
</tbody>
</table>

5. Save changes.

Now, we must add an event that will dynamically set Iframe URL depending on the action triggered. There are two possible actions - new and edit. To do so, follow next steps.

1. Go to Settings ➤ Customizations ➤ Customize the System ➤ .
2. Locate the CPQ Quote entity at the left pane, then click on Forms submenu, then click on Information (Main) form to edit it.
3. Under HOME tab click on Form Properties. Form Properties dialog box is displayed.
4. Click **Add** under **Form Libraries** area. This opens another dialog box - **Web Resource**, where you add new Resource (Form Library). To do so, click **New**. This opens a dialog box where you define new Script (JavaScript). Fill out fields as instructed:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Field value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>SetUrlForCpqQuote</td>
<td><strong>new_</strong> is prepended automatically</td>
</tr>
<tr>
<td>Type</td>
<td>Script (JScript)</td>
<td><strong>new_</strong> is prepended automatically</td>
</tr>
</tbody>
</table>

5. Click **Text Editor** to paste JavaScript code that is provided below into the Script editor. Note that the only thing you should modify in the script is SAP CPQ domain name.

```javascript
function SetUrlForCpqQuote() {
  opp = Xrm.Page.data.entity.attributes.get("new_opportunity").getValue();
  var userId = Xrm.Page.context.getUserId();
  var oppId = opp[0].id;
  var newquoteurl = "https://sandbox.webcomcpq.com/sso/mscrm.aspx?domain_name=sap&userId="+userId+'&typename=opportunity&id='+oppId;
  linktoquote = Xrm.Page.data.entity.attributes.get("new_linktoquote").getValue();
  var IFrame = Xrm.Page.ui.controls.get("IFRAME_CPQ");
  if (linktoquote) {
    IFrame.setSrc(linktoquote);
  } else {
    IFrame.setSrc(newquoteurl);
  }
}
```

Click **OK**.

6. Now you can add the Form Library you have just created. To do so, while you are in **Form Properties** dialog box, click **Add** under **Form Library** area, then search for the library and click **Add**.

7. Now, add new **Event Handler** by clicking on **Add** under **Event Handler** area. This opens a new dialog box - **Handler Properties**. In this dialog enter name of the function (SetUrlForCpqQuote)

**Step 5: Navigating back to Opportunity**

You can navigate back to the Opportunity form by clicking on **Recently Viewed** Items from top navigation bar, but the more convenient way is to click on button. This button must be added as a Web Resource first. Follow next steps to create it.

1. Go to **Settings** ➤ **Customizations** ➤ **Customize the System** ➤
2. Locate the **CPQ Quote** entity at the left pane, then click on **Forms** submenu, then click on **Information (Main)** form to edit it.
3. On next screen, navigate to the **INSERT** tab, then select area in section CPQ Quote, and finally click **Web Resource** button.
4. **Add Web Resource** dialog box is open. Click on **Search** icon and then click on **Look up More Records**.
5. New dialog box - **Look Up Record** is open. Click **New**.
6. New dialog box for adding new Web Resource is open. Fill out fields as instructed:
Click Text Editor button to open text editor. Click on Source tab to open it. Copy and paste code provided here and then click OK.

```html
<html><head>
<title>Back to Opportunity</title>
<meta></script>
function BackToOpportunity() {
    var opp = window.parent.Xrm.Page.data.entity.attributes.get("new_opportunity").getValue();
    var oppId = opp[0].id;
}
</script>
<meta></head>
<body style="word-wrap: break-word;">
<input type="button" value="Back to Opportunity" onclick="BackToOpportunity();">
</body></html>
```

Web Resource details are displayed in the dialog box. You are done with creating Web Resource. Click OK. Web Resource appears in Opportunity Quotes section.

Save SAP CPQ Quote form and publish customizations.

### SAP CPQ Setup

**Step 1: Define general parameters**

Beside CRM, you must do some setup steps on SAP CPQ side too. This portion of text describes in detail what you should set in order to make your CRM - SAP CPQ integration operable.

Log into your SAP CPQ domain and go to Setup > CRM Integration > General > Under CPQ is integrated with drop-down list choose Microsoft CRM.

Under CRM Webservice Discovery URL enter:

For CRM Online customers:
- For North America: https://dev.crm.dynamics.com/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx
- For EMEA: https://dev.crm4.dynamics.com/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx
- For APAC: https://dev.crm5.dynamics.com/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx

For CRM On-premise customers:
- If you are using SPLA authentication: https://{yourservername}/MSCRMServices/2007/SPLA/CrmDiscoveryService.asmx
If you are using Passport authentication: https://{yourservername}/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx
If you are using Active Directory authentication: https://{yourservername}/MSCRMServices/2007/AD/CrmDiscoveryService.asmx

Under Your company name in CRM, enter your company name.

Under Domain name (enter if using Active Directory), enter your domain name only if you are using Active Directory.

Under Organization URL enter: https://{YourCompanyName}.crm.dynamics.com/XRMServices/2011/Organization.svc

Under Endpoint type enter LiveId

**Step 2: Define an administrative account**

Now you must define an administrative account. To do so, follow next steps.

- Go to Setup CRM Integration CRM Administrator Account.
- Click Add New to create a new administrator account.

In the displayed form, choose Microsoft CRM under CRM Name, then fill out other fields based on your data and needs and click Save.

**Step 3: Define custom actions**

New custom actions should be defined to provide a way to update opportunity in CRM through SAP CPQ, and to move the opportunity from one stage to another. Custom actions are programmed using Iron Python. Follow next steps to add new custom action.

- Go to Setup Develop Develop Custom Actions.
- Click Add New fill out the form with necessary data (action name, action image, placement...)

A list of custom actions, developed using Iron Python, can be found on this link: Custom Actions.

**Step 4: Set Workflow and Workflow permission**

The last step is dependent on your model and it assumes that you must set Workflow permission for users or user types as desired, and set Workflow transitions. To find more details on how to do this, check this link: Workflow Permissions.

**Related Information**

MS Dynamics CRM (2011 version) and SAP CPQ Integration Instructions [page 251]
Dynamics CRM Integration Instructions for MS CRM 4.0 [page 253]
7.2 MS Dynamics CRM (2011 version) and SAP CPQ Integration Instructions

SAP CPQ can be integrated with MS Dynamics CRM (2011 version).

CRM Setup

**Note**

All mapping of entities in the two systems should be handled via IronPython scripting, and not in the SAP CPQ Setup.

In order to access SAP CPQ Quotes through MS CRM opportunity, you need to log in to MS CRM with administrative account. After you have logged in follow these steps:

- click on Settings
- in the left menu click on Customizations
- click on Customize the System
- Among entities find the Opportunity object and click on it
- Click on Forms and then find the form that you want to edit (one used in opportunity)

You can define SAP CPQ as either Iframe that will be displayed on opportunity page in particular section, or as a new tab where Iframe with SAP CPQ will be displayed. In order to add Iframe either to tab or to section, click on IFRAME under the Insert tab. Add an IFRAME form is displayed. Follow next steps to set up an SAP CPQ IFRAME.

- Under name field write **CPQ**.
- Under URL field write **https://CpqApplicationName/sso/mscrm.aspx?domain_name=DomainName**
  Replace CpqApplicationName and DomainName with the CPQ instance URL (for example www.webcomcpq.com) and with the name of your domain, respectively.
- Under label field type **CPQ**
- Be sure to check **Pass record object-type code and unique identifier as parameters**
- Under Security, uncheck **Restrict cross-frame scripting**.
- Click on tab Formatting and set the number of rows to 35. Make sure that checkbox **Automatically expand to use available space** is selected.
- Save Iframe.

Click on Form Properties.

- On tab Events, under Event List click on Add
- Under Name, put CPQScript
- For Type, pick Script(JScript)
- Click on Text Editor, and add following code:

```javascript
function SetUrlForCpq()
{
    var IFrame = Xrm.Page.ui.controls.get("IFRAME_CPQ");
    var Url = IFrame.getSrc() + ":userId=" + Xrm.Page.context.getUserId();
}```
Iframe.setSrc(Url);
}

Save the script and publish it. At the screen top, under dropdown Actions, click on Publish.

On the same tab Events, under section Event Handlers, for Control : Form and Event : OnLoad, click on Add and for Library select new_CPQScript. Leave flag Enabled selected and enter SetUrlForCpq for Function, since that is the function from our script. Click OK. Save form and publish it.

**SAP CPQ Setup**

Log into your SAP CPQ domain and go to Setup > CRM Integration > General > Under CPQ is integrated with dropdown list choose Microsoft CRM. Under CRM Webservice Discovery URL enter:

For CRM Online customers:
- For North America: https://dev.crm.dynamics.com/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx
- For EMEA: https://dev.crm4.dynamics.com/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx
- For APAC: https://dev.crm5.dynamics.com/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx

For CRM On-premise customers:
- If you are using SPLA authentication: https://{yourservername}/MSCRMServices/2007/SPLA/CrmDiscoveryService.asmx
- If you are using Passport authentication: https://{yourservername}/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx
- If you are using Active Directory authentication: https://{yourservername}/MSCRMServices/2007/AD/CrmDiscoveryService.asmx

Under **Your company name in CRM**, enter your company name.

Under **Domain name (enter if using Active Directory)**, enter your domain name only if you are using Active Directory.

Under **Organization URI** enter: https://{YourCompanyName}.crm.dynamics.com/XRMServices/2011/Organization.svc

Under **Endpoint type** enter LiveId.

**Define an administrative account**

Now you must define an administrative account. To do so, follow next steps.

- Go to Setup > CRM Integration > CRM Administrator Account
- Click Add New to create a new administrator account.

In the form, choose Microsoft CRM under **CRM Name**, then fill out other fields based on your data and needs and click Save.
Dynamics CRM Integration using IronPython scripting

You can find instructions on how to integrate SAP CPQ with MS CRM on the following help page: Integrate with MS CRM.

Related Information

MS Dynamics CRM and SAP CPQ Integration [page 244]
Dynamics CRM Integration Instructions for MS CRM 4.0 [page 253]

7.3 Dynamics CRM Integration Instructions for MS CRM 4.0

In order to access Cpq Quotes through MS CRM opportunity, you need to log in to MS CRM with administrative account.

After you have logged in, click on Settings, then in the left menu click on Customization ➔ Customization Entities ➔ and among entities find the Opportunity object and click on it. When the window is opened, on the left, click on Forms and Views, and then among All Forms and Views, find Form (Name = Form and Type = Form) and click on it.

On the right, click on Add a Tab, and name the tab ‘Cpq Quotes’. For the new tab click on Add a Section, name it ‘CpqSection’ and save it.

Click on Add an IFRAME.

- Under name, in the second field write ‘CPQ’.
- For the URL write: https://CpqApplicationName/sso/mscrm.aspx?domain_name=DomainName (replace CpqApplicationName and DomainName with the URL of your Cpq application URL (for instance v6.webcomcpq.com) and with the name of your domain, respectively).
- Be sure to check Pass record object-type code and unique identifier as parameters
- Under Security, uncheck Restrict cross-frame scripting.
- Click on tab Formatting. Make sure that checkbox Automatically expand to use available space is selected.
- Save iframe.

On the right, click on Form Properties.

- Under Events, choose OnLoad and click Edit
- Check Event is enabled and it the text area paste the following text:

```javascript
var xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
xmlhttp.open("POST", "/mscrmservices/2007/crmservice.asmx", false);
xmlhttp.setRequestHeader("Content-Type", "text/xml; charset=utf-8");

var soapBody = "<soap:Body">"+
```
Save and close the Form. At the screen top, under dropdown Actions, click on Publish.

**Related Information**

MS Dynamics CRM (2011 version) and SAP CPQ Integration Instructions [page 251]
MS Dynamics CRM and SAP CPQ Integration [page 244]
8  Salesforce Integration

This section contains instructions and useful information for integrating SAP CPQ with SalesForce.

- How to set up account look-ups from SAP CPQ customer page [page 255]
- New Opportunity Save Behavior [page 256]
- Salesforce Chatter Integration [page 256]
  - SAP CPQ is integrated with Salesforce Chatter which enables SAP CPQ to post quotes comment to Salesforce Account or Opportunity chatter.
- SAP CPQ Package for SalesForce [page 257]
  - The SAP CPQ Package for Salesforce integration enables you to automatize the integration process with minimal additional setup adjustments.

8.1  How to set up account look-ups from SAP CPQ customer page

1. Make sure that Salesforce administrator account is set up in SAP CPQ.
   This is required when users log in directly to SAP CPQ instead of arriving from Salesforce. To see how to set up a Salesforce administrator account, please go to CRM Administrator Account [page 288].
2. Set up mapping between SAP CPQ companies and accounts in Salesforce.
   When user logs in directly to SAP CPQ and performs account look-up from SAP CPQ SAP CPQ customer page, user will be able to see only their company account, its child accounts and its partner accounts. In order to set up user’s company account, you need to find user’s company in SAP CPQ Setup (Companies), and in field ‘CRM Account ID’ to enter account ID from Salesforce that presents user’s company.
3. Make sure that action ‘CRM Lookup’ is enabled for appropriate SAP CPQ customer roles
   This is done in SAP CPQ → Setup → Customers/Customer Roles → Customer Actions →
4. Define partner roles that will be visible to user when performing account look-up.
   When users perform look-ups from customer page, they can see partner accounts of their company account. You can define which partner accounts users are able to see (e.g. Resellers, Partners, etc.) in SAP CPQ → Setup → CRM Integration Setup → General under the parameter ‘Partner Role Type’.

In order to see how look-up screen looks like, please go to CRM Integration Use.

Related Information

- Salesforce Integration [page 255]
8.2  New Opportunity Save Behavior

In their Spring '09 release, Salesforce has introduced new opportunity save behavior feature. It was published only to sandbox installations at the time, however the new feature will be enforced to all Salesforce applications as of December 2009.

The purpose of this feature is change the save behavior for Opportunities so that all standard and custom save logic is consistently fired, aligning the behavior across the Force.com Platform.

New opportunity save behavior may impact your Salesforce - SAP CPQ integration. Most common scenario may be if you have opportunity validation rules set up in your Salesforce application that check whether opportunity total is bigger than required minimal amount. When pushing quote information to Salesforce opportunity, SAP CPQ first deletes existing opportunity products and then recreates them with new date found in quote items. The change that comes with new opportunity save behavior is reflected at the start of the process - previously, deleting opportunity products did not trigger opportunity validation rules, while with new release opportunity validation rules are executed at this point and may return the error.

**Note**

To read more about opportunity save behavior, please go to [New Opportunity Save Behavior](#).

If your Salesforce - SAP CPQ integration is affected with this change, recommended steps for resolving the issue are:

1. Go to Salesforce Setup and create opportunity custom field (type checkbox). Default for this field should be false. When updating opportunities, SAP CPQ will set this field to ‘TRUE’.

2. In SAP CPQ [Setup > CRM Integration Setup > General], there is an entry 'Opportunity Validation flag'. Click on ‘CRM Lookup’ link and find the API name of the opportunity field that was created in step 1.

3. Modify opportunity validation rules so that critical validations are implemented only when this flag is set to FALSE (purpose of the change is to distinguish situations where SAP CPQ is updating the opportunity vs. when it is user who is changing it).

**Related Information**

[Salesforce Integration](#) [page 255]

8.3  Salesforce Chatter Integration

SAP CPQ is integrated with Salesforce Chatter which enables SAP CPQ to post quotes comment to Salesforce Account or Opportunity chatter.

For this purpose two standard quote actions can be found under [Setup > Workflow/Approval > Actions]:

- Post Quote Notes Into Opportunity Chatter
• **Post Quote Notes Into Account Chatter**

These two actions can be set in Workflow as a post action to any other standard SAP CPQ action, based on your needs. Usually, one or both actions are set as post actions for `CreateUpdateOpportunity` SAP CPQ action, that guarantee that quote comments will be refreshed in Salesforce after every Opportunity update or creation.

To set a post action for `CreateUpdateOpportunity` action go to **Setup** ➔ **Workflow/Approval** ➔ **Workflow** ➔ In the Workflow matrix, find the desired Start and End status intersection and click the edit button.

On the next page, select the `Create Update Opportunity` action and click the edit button in the **Post Actions** column to add the post action.

On the next screen, from the list of available actions, choose the desired one (for example, **Post Quote Notes into Account Chatter**) and click **Update**. Click **Save**.

When you are on quote, each time the action `Create Update Opportunity` is triggered, the post action is triggered as well, and comment posted to the Salesforce account chatter.

### Related Information

Salesforce Integration [page 255]

### 8.4 SAP CPQ Package for Salesforce

The SAP CPQ Package for Salesforce integration enables you to automatize the integration process with minimal additional setup adjustments.

The advantages of integrating SAP CPQ and Salesforce via the package are the following:

- Quote creation is optimized. After you create an opportunity in Salesforce and land to SAP CPQ by clicking **New Quote**, you are able to create a new quote and update the opportunity by slicking **Create/Update Opportunity**.
- User provisioning is enabled. The first time you land to SAP CPQ a matching user is automatically created. The only prerequisite is your Salesforce system administrator having a CRM administrator account in SAP CPQ.
- Simple and straightforward UI of the integrated environment provides streamlined opportunity/quote management.
- Single Sign-On authentication service that allows users to use one password and username to access both Salesforce and SAP CPQ.

**i Note**

Integration of SAP CPQ and Salesforce is supported by both Quote 1.0 and Quote 2.0.

To integrate Salesforce and SAP CPQ instances, you must combine two production or two test instances:

- **https://test.salesforce.com**- **https://sandbox.webcom.com**
- **https://salesforce.com**- **https://rs.webcomcpq.com** and **https://v60test.webcomcpq.com**
• [https://na73.salesforce.com](https://na73.salesforce.com) (development instance, the same version as the production environment) - [https://v60test.webcomcpq.com](https://v60test.webcomcpq.com)

There are more SAP CPQ environments that can be integrated with Salesforce and they are listed in the table:

<table>
<thead>
<tr>
<th>Production</th>
<th>Sandbox</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://www.webcomcpq.com">https://www.webcomcpq.com</a></td>
<td><a href="https://sandbox2.webcomcpq.com">https://sandbox2.webcomcpq.com</a></td>
<td><a href="https://teststable.webcomcpq.com">https://teststable.webcomcpq.com</a></td>
</tr>
<tr>
<td><a href="https://rs.webcomcpq.com">https://rs.webcomcpq.com</a></td>
<td><a href="https://eusb.webcomcpq.com">https://eusb.webcomcpq.com</a></td>
<td><a href="https://v60test.webcomcpq.com">https://v60test.webcomcpq.com</a></td>
</tr>
<tr>
<td><a href="https://www1.webcomcpq.com">https://www1.webcomcpq.com</a></td>
<td><a href="https://eutest.webcomcpq.com">https://eutest.webcomcpq.com</a></td>
<td><a href="https://eu.webcomcpq.com">https://eu.webcomcpq.com</a></td>
</tr>
</tbody>
</table>

**iNote**

Package integration is tenant-specific (an account in SalesForce corresponds to a tenant in SAP CPQ). Once a package is downloaded and installed and the adjustments in the Setup are finished, the integration is enabled for all users in the tenant. You can always limit the access to all users or to specific profiles in AppExchange. Since the integration requires certain adjustments in Setup, only administrators can integrate SAP CPQ and Salesforce properly via the package.

### Download the Package

**Prerequisite:** to have an administrator account created in [Salesforce](https://www.salesforce.com).

For the purposes of the procedures, we will integrate the Salesforce development environment [https://na73.salesforce.com](https://na73.salesforce.com) with SAP CPQ test environment [https://v60test.webcomcpq.com](https://v60test.webcomcpq.com).

1. Log into Salesforce as an administrator. You can use the following link to log in and download the package, which you are prompted to do the first time you access the page. In the future, if a new version of the package is available, the system prompts you to upgrade to the new version. A page with three installation options displays:
   - **Install for Admins Only** - enables the integration only for Salesforce administrators.
   - **Install for All Users** - enables the integration for both administrators and their underlying users.
   - **Install for Specific Profiles...** - enables the integration for the selected profiles.
2. Select **Install for All Users**. The integration is now enabled for all your underlying users.
3. Click **Install**. When upgrading to a new version, click **Upgrade**.
4. Click **Done** when the process is completed.
   The package is now installed and you are redirected to the Salesforce home page. After you make additional adjustments in Setup, all your users will be able to use the SAP CPQ-Salesforce integrated environment.
You can check if the package is installed in Setup ➤ Build ➤ Installed Packages.

**Setup Adjustments - SalesForce side**

After you download the package, you should make additional adjustments in SalesForce Setup.

**Note**

The procedures are applicable to the Salesforce Classic UI design.

**Manage Access to Connected Apps**

Follow the procedure to limit the access to the application only to users with the appropriate profile or permission set.

1. In **Administer**, access Manage Apps ➤ Connected Apps.
   
   All SAP CPQ instances that can be integrated with SalesForce are displayed.

2. Click **Edit** next to **CPQOAuth2_v60Test**.
   
   In OAuth policies ➤ Permitted Users, select **Admin approved users are pre-authorized**.

3. Click **Ok**.

4. Click **Save**.
   
   The confirmation pop-up no longer displays to users that are preauthorized.

**Allow User Profiles to use CPQOAuth 2.0**

OAuth 2.0 authorization framework simplifies all the communications between the client, server, and content provider. Follow the steps to enable a profile to use OAuth 2.0.

1. In **Administer**, access Manage Users ➤ Profiles.
   
   All user profiles in the system display.

2. Click **Edit** next to the profile you wish to edit.
   
   A page where you can edit profile permissions and the page layout displays.

3. In **Connected App Access** select **CPQOAuth2_v60Test**.

4. Click **Save**.
   
   The user profile is now able to use CPQOAuth 2.0 authorization.

**Note**

The user profile is selected while creating the user. You can always double check it in **Administer** ➤ Manage Users ➤ Users in the Profile column. Additionally, for a successful integration, the same user type needs to exist in SAP CPQ.

**Define Custom Settings**

Custom settings let you utilize custom data sets across your organization. In addition, custom settings allow distinguishing particular users or profiles based on custom criteria.
1. In Build, access Develop Custom Settings.

2. Click Manage next to CPQ Settings.

3. Click New.

4. Enter the data in the dedicated fields:
   - **Name** - Name of the data set (the name is not conditioned by other settings. You can enter CPQ, for example).
   - **TenantKey** - Automatically populated by the system. Not a required field.
   - **TenantName** - Name of the SAP CPQ domain.

5. Click Save. The custom settings are now defined.

Add Quote to Opportunity Level

In order to access Quotes from Opportunities in SalesForce, you need to make adjustments of the SalesForce user interface.

1. In Build, access Customize Opportunities.

2. Click Page Layouts. All opportunity page layouts in the system display.

3. Click Edit next to the opportunity layout you wish to edit. Not all layouts are available for all user profiles.

4. In the Opportunity Layout pallet, in the list on the left, click Related Lists.

5. Drag Quotes from the pallet and drop it into Opportunity Sample under Related Lists.

6. Click Save in the pallet. The user interface is now adjusted and you can move from opportunities to quotes.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>In case the customization differs from the users' personal customization, the Overwrite Users' Related List Customizations? pop-up will display. Click Yes to apply your customization.</td>
</tr>
</tbody>
</table>

Add Company and Brand to User Layout

For a successful integration, the user in SalesForce should have the company and the brand available in SAP CPQ.

1. In Build, access Customize Users.

2. Click Page Layouts. All user page and user profile page layouts display.

3. In User Page Layouts, click Edit next to User Layout.

4. From the pallet, drag Brand and Company Name and drop them into Additional Information.

5. Click Save in the pallet.

6. In Administer, access Manage Users Users.

7. Click Edit next to the user whose data you wish to edit.

8. Enter the company name and the brand in Additional Information.

9. Click Save. The brand and the company of the user are defined.
Setup Adjustments – SAP CPQ side

To integrate SAP CPQ with SalesForce successfully, you should log into SAP CPQ using the tenant that will be integrated and make additional adjustments in Setup. All SalesForce users can land to SAP CPQ as long as their SalesForce system administrator has a CRM administrator account in SAP CPQ. A CRM administrator account can be created in CRM Administrator Account [page 288] part of Setup.

General Adjustments

1. Copy the domain secret key from Salesforce and define the value of two general parameters.
   The domain secret key is available in Salesforce under Build ➤ Develop ➤ Custom Settings ➤ Manage ➤ Edit ➤ TenantKey.
2. Access CRM Integration ➤ General ➤.
3. In the CPQ is integrated with list, select SalesForce.
4. Click Save.
   The page and the menu are refreshed.
5. Enter the value in Domain secret key.
6. Update the parameters.
7. Remove the value from Security password for verifying entrance to SAP CPQ through CRM backdoor.
   This parameter needs to be empty.
8. In CRM Quote Opportunity API Name (SF Package Only) enter the following value:
   v60test__Opportunity__c
   Please note that the value is the same for test, sandbox and production environments. Make sure to copy/paste the value to avoid any typos.
9. Set Allow SF Package Integration to True if not set by default.
10. Click Save.

Add Quote object names

Map the opportunity and quote fields so you can create a quote from the opportunity.

1. Access CRM Integration ➤ CRM Quote ➤.
2. Fill out the fields with the following values.
   ○ CRM Quote Object Name:
     v60test__Quote__c
   ○ CRM Field For Persisting Quote Id:
     v60test__Quote_Id__c
   ○ CRM Field For Persisting Quote Owner Id:
     v60test__Owner_Id__c
   ○ CRM Field For Persisting Information About Primary Quote:
     v60test__Primary__c
3. Click Save.
   You are now able to create a quote from the SalesForce opportunity.
The same value should be assigned to the fields regardless of which SAP CPQ environment (test, sandbox and production) is being integrated. Please make sure to copy/paste the values to avoid any typos.

Once the setup adjustments are completed, you can start using the integrated SalesForce - SAP CPQ environment. When you create an opportunity and click *New Quote*, the system redirects you to SAP CPQ. The first time you land to SAP CPQ, a matching user is automatically created, no additional setup is needed. Clicking *Create/Update Opportunity* on the quote updates the existing opportunity in SalesForce.

The product should have a part number for the system to successfully create a quote from an opportunity.

**Related Information**

- Salesforce Package Integration Troubleshooting
- Salesforce Integration [page 255]
9 Salesforce Integration Instructions

This guide is to help you set up the integration between SAP CPQ and Salesforce.com (SFDC) so an end-user can create quotes within the SalesForce environment as well as populate the quote back into the Salesforce sales opportunity.

If you haven’t already, you need to create a developers account at Salesforce. Go to the following website and sign up: https://developer.salesforce.com

You will also need your SAP CPQ tenant info. If you are embarking on this setup, you most likely have a tenant. If not, you should contact your manager to get a tenant.

_**i Note**_

For successful integration, you should enable cookies in the browser, whether on your computer or the mobile device.

**Salesforce Setup**

There are 6 essential steps to set up the SAP CPQ - Salesforce integration. Read over the concepts first as they will assist you in the task.

_**i Note**_

Integration of SAP CPQ and Salesforce is supported by both Quote 1.0 and Quote 2.0.

First, you need to create a custom object in Salesforce for the quote.

**Prerequisites**

- Log into the Salesforce site.
- Before you start, know where to enter the setup feature in Salesforce. At any time you can click the Setup button to get back into setup. Setup is found in the upper right-hand side of Salesforce.
- Follow the directions closely, they will save you valuable time in the long run.

_**i Note**_

- Most of the parameters are case sensitive.
- Salesforce uses __c at the end of their objects, this is a double underscore!
- Salesforce is constantly upgrading their solution, so the below directions can easily be different in the tenant you are logged into, the procedures may be the same, however, but the menus may have changed.
- There are sometimes a few different ways to get to something you’re working on.
- Remember to click Save at the end of each task.
Step 1: Create custom object "Quote"

- Click on Setup link at the top-right menu.
- Click on Add App button.
- Fill out the input fields with the following data:
  - in App field type Quote
  - in Label field type Quote
  - in Plural Label field type Quotes

You can access the quote object via menu | Build | Create | Objects |. List of custom objects is displayed in a table. Click on Quote link to open Custom Object Details page. This page shows all properties related to your custom object Quote.

Step 2: Add Custom Fields to Quote Object

Now that you defined the Quote object you need to define the fields in Salesforce that we will eventually match up to SAP CPQ. To add custom fields, click on new link under Custom Fields & Relationships submenu from the top menu. You will need to create eight fields with the parameters from the table to the Quote object.

⚠️ Caution

Fields are case-sensitive and remember, the _ before letter “c” (_c) is a double underscore!

Field names and attributes that you must create are:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discount Percent</td>
<td>Percent</td>
<td>Length - 16, Decimal Places – 2</td>
</tr>
<tr>
<td>Total List Price</td>
<td>Currency</td>
<td>Length - 16, Decimal Places – 2</td>
</tr>
<tr>
<td>Opportunity</td>
<td>Lookup Relationship</td>
<td>Related to Opportunity</td>
</tr>
<tr>
<td>Owner ID</td>
<td>Number</td>
<td>Length - 18, Decimal Places – 0</td>
</tr>
<tr>
<td>Primary</td>
<td>Checkbox</td>
<td></td>
</tr>
<tr>
<td>Quote ID</td>
<td>Number</td>
<td>Length - 18, Decimal Places - 0</td>
</tr>
<tr>
<td>Total Net Price</td>
<td>Currency</td>
<td>Length - 16, Decimal Places – 2</td>
</tr>
<tr>
<td>Revision Number</td>
<td>Text</td>
<td>10</td>
</tr>
</tbody>
</table>

Let’s walk through the first one and then you can do all other fields with the same procedure.

Example for field Discount Percent:

- Select the field type.
- Click Next and fill out the following fields: Field Label, Length, Decimal Places, Field Name, Help Text.
- Click Next. Custom field-level security options are displayed. On this page, you can define which Profiles will be able to see your field and which not.
- Click Next. On this page, you can define Layout page where your custom field will be displayed. If quote Layout option is not checked, please check it and click Save. You can click Save & New if you are ready to
create the next field. Once you have finished creating all eight fields from the table, you’re ready for the next step.

- You are done with custom fields definitions. You can find a list of custom fields by going to Build Objects and clicking on quote label.
- Now, under Standard fields section, find quote Name standard field and click on edit next to field name. Rename it from quote Name to Quote Number.

Step 3: Create a Visualforce Page

You now need to create a Visualforce (VF) Page. Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Force.com platform. The Visualforce framework includes a tag-based markup language, similar to HTML, and a set of server-side “standard controllers” that make basic database operations, such as queries and saves, very simple to perform.

- Go to Setup Build Develop Visualforce Pages.
- First, create new view page by clicking on Create New View link.
- View creation form is displayed. Fill out fields this way:
  - View Name: CPQ Installation
- In Step 4 section check radio button with label Visible to all users (included partner and customer portal users). This will make a view visible to all.
- Click Save.

Now you will create two pages - one for creating a new Quote, and another for editing a Quote. To create a new page, click on New button. First, create a New quote:

- Label: CPQQuoteNew
- Name: CPQQuoteNew

i Note

Select the Available for Lightning Experience, Lightning Communities, and the mobile app checkbox. This makes the page available in the lightning design and on mobile devices.

Cut and Paste the apex code provided below, and paste into the text box under Visualforce Markup tab. You must replace the URL with the URL of the SAP CPQ website you want to connect to, as well as the domain name.

```apex
<apex:page standardController=”Quote__c” showHeader=”true”>
<script type=”text/javascript”>
    document.onready = function () {
        var iframeElm = document.getElementById(‘iframe1’);
        var parentStyle = window.getComputedStyle(iframeElm.offsetParent);
        var parentPaddingTop = parseInt(parentStyle.paddingTop);
        var parentPaddingBottom = parseInt(parentStyle.paddingBottom);
        var iframeBorder = parseInt(window.getComputedStyle(iframeElm).borderWidth) || 2;
        var iframeParentPadding = parentPaddingTop + parentPaddingBottom;
        // only do if on mobile
        if((typeof sforce != ‘undefined’) && sforce && (!!sforce.one)){
            // when document is ready set height, width and style
            setHeight(iframeElm, iframeParentPadding + iframeBorder);
            setWidth(iframeElm, iframeBorder);
        }
    }
</script>
```
window.addEventListener('resize', function() {
  setHeight(iframeElm, iframeParentPadding, iframeBorder);
  setWidth(iframeElm, iframeBorder);
}, true);
else {
  setClassicHeight(iframeParentPadding, iframeBorder);
  window.addEventListener('resize', function() {
    setClassicHeight(iframeParentPadding, iframeBorder);
  }, true);
}

function setClassicHeight(iframeParentPadding, iframeBorder) {
  var iframeEl = document.getElementById('iframe1');
  var headerEl = document.getElementById('AppBodyHeader');
  var footerEl = document.getElementsByClassName('bPageFooter')[0];
  var tableEl = document.getElementsByClassName('bodyDiv')[0];
  // Get height values for all Salesforce elements (header, footer and container padding)
  var sfElementsHeight = parseInt(headerEl.offsetHeight) +
    parseInt(footerEl.offsetHeight) +
    parseInt(window.getComputedStyle(tableEl).borderTopWidth);
  // Set minimum iframe height
  iframeEl.style.minHeight = '600px';
  iframeEl.style.height = window.innerHeight - sfElementsHeight - iframeParentPadding - (2 * iframeBorder) + 'px';
}

function setHeight(iframeElm, iframeParentPadding, iframeBorder) {
  // height is set to full window
  var newHeight = window.innerHeight - iframeParentPadding - (2 * iframeBorder) + 'px';
  iframeElm.style.height = newHeight;
  // if not already set set maxHeight to iframe url so that SAP CPQ sets correct size
  if (iframeElm.src.indexOf("maxHeight") === -1) {
    iframeElm.src = iframeElm.src + '&maxHeight=' + newHeight;
  }
}

function setWidth(iframeElm, iframeBorder) {
  // Set width to parent width - 2 * border size to avoid horizontal scroll
  var widthValue = document.body.clientWidth - (2 * iframeBorder) + 'px';
  iframeElm.style.width = widthValue;
}

function setStyle(iframeElm){
  // Get style of parent element
  var parentStyle = window.getComputedStyle(iframeElm.offsetParent);
  var parentLeft = parentStyle.paddingLeft;
  var parentRight = parentStyle.paddingRight;

  // setting margin and overflow
  iframeElm.style.marginLeft = '-' + parentLeft;
  iframeElm.style.marginRight = '-' + parentRight;
  iframeElm.style.overflow = "hidden";
}
</script>
Now create the Edit quote.

- **Label**: CPQQuoteEdit
- **Name**: CPQQuoteEdit

Select the **Available for Lightning Experience, Lightning Communities, and the mobile app** checkbox. This makes the page available in the lightning design and on mobile devices.

In the Visualforce Markup tab, add the following code. Again, replace the URL and the domain name.

```html
<apex:page standardController="Quote__c" showHeader="true" >
<script type="text/javascript">
  document.onreadystatechange = function () {
    var iframeElm = document.getElementById('iframe1');
    var parentStyle = window.getComputedStyle(iframeElm.offsetParent);
    var parentPaddingTop = parseInt(parentStyle.paddingTop);
    var parentPaddingBottom = parseInt(parentStyle.paddingBottom);
    var iframeBorder = parseInt(window.getComputedStyle(iframeElm).borderWidth) || 2;
    var iframeParentPadding = parentPaddingTop + parentPaddingBottom;

    // only do if on mobile
    if((typeof sforce != 'undefined') && sforce && (!!sforce.one)){
      // when document is ready set height, width and style
      setHeight(iframeElm, iframeParentPadding, iframeBorder);
      setWidth(iframeElm, iframeBorder);
      setStyle(iframeElm);

      // if window is resized adjust width again to be responsive
      window.addEventListener("resize", function () {
        setHeight(iframeElm, iframeParentPadding, iframeBorder);
        setWidth(iframeElm, iframeBorder);
        }, true);
    } else {
      setClasicHeight(iframeParentPadding, iframeBorder);
      window.addEventListener("resize", function () {
        setClasicHeight(iframeParentPadding, iframeBorder);
        }, true);
    }
  }

  function setClasicHeight(iframeParentPadding , iframeBorder) {
    var iframeEl = document.getElementById('iframe1');
    var headerEl = document.getElementById('AppBodyHeader');
    var footerEl = document.getElementsByClassName('bPageFooter')[0];
    var tableEl = document.getElementsByClassName('bodyDiv')[0];

    // Get height values for all Salesforce elements (header, footer and container padding)
    var sfElementsHeight = parseInt(headerEl.offsetHeight) +
                           parseInt(footerEl.offsetHeight) +
                           parseInt(window.getComputedStyle(tableEl).borderTopWidth);

    // Set minimum iframe height
    iframeEl.style.minHeight = '600px';
    iframeEl.style.height = window.innerHeight - sfElementsHeight - iframeParentPadding - (2 * iframeBorder) + "px";
  }

  function setHeight(iframeElm, iframeParentPadding, iframeBorder) {
    // height is set to full window
    var newHeight = window.innerHeight - iframeParentPadding - (2 * iframeBorder) + "px";
    iframeElm.style.height = newHeight;
  }
</script>
</apex:page>
```
// if not already set set maxHeight to iframe url so that SAP CPQ sets correct size
if (iframeElm.src.indexOf("maxHeight") === -1) {
    iframeElm.src = iframeElm.src + "&maxHeight=" + newHeight;
}
}

function setWidth(iframeElm, iframeBorder) {
    // Set width to parent width - 2 * border size to avoid horizontal scroll
    var widthValue = document.body.clientWidth - (2 * iframeBorder) + "px";
    iframeElm.style.width = widthValue;
}

function setStyle(iframeElm) {
    // Get style of parent element
    var parentStyle = window.getComputedStyle(iframeElm.offsetParent);
    var parentLeft = parentStyle.paddingLeft;
    var parentRight = parentStyle.paddingRight;

    // setting margin and overflow
    iframeElm.style.marginLeft = '-' + parentLeft;
    iframeElm.style.marginRight = '-' + parentRight;
    iframeElm.style.overflow = "hidden";
}
</script>
<iframe id="iframe1" width="100%" scrolling="true" src="https://sandbox.webcomcpq.com/salesforce/SfLogin.aspx?sfauthUserID={!$User.Username}&apiPartnerURL={!$Api.Partner_Server_URL_290}&apiSessionID={!$Api.Session_ID}&domain_name=TestDomain&apiPass=Password&action=Edit&sfqpOpportunityID={!Quote__c.Opportunity__c}&OwnerId={!CEILING(Quote__c.Owner_Id__c)}&QuoteId={!CEILING(Quote__c.Quote_Id__c)}"></iframe>

**Quote Cloning - Optional**

Quote Cloning is not required for setting up the initial integration with Salesforce.

To carry out Quote Cloning: Create S-Control (due to the error that exists in Salesforce Visual Force page functionality, we are not able to provide instructions how to set up this feature by using the VF page. As soon as the error is resolved in Salesforce, we will post instructions here).

Label: Quote - Clone Primary.

S-Control Name: Quote_Clone_Primary

Type: HTML

Content:

<script>var me = window.name;
if (me){
    var iframes = parent.document.getElementsByTagName('iframe');
    if (iframes &amp; iframes.length == 1) {
        iframes[0].style.height = 1000 + "px";
    }
}

In the Custom Buttons and Links section of the new custom object, you should add a new button:
You can also use apiPass to enhance security. In CRM Integration Setup General, there is an entry Security password for verifying entrance to CPQ through the Salesforce backdoor. Use the word “password” as your password.

If this entry is left blank, apiPass from the Visual Force page URL will not be verified upon entrance through the backdoor. However, if this entry is not empty the value written in the Visual Force page URL will be matched with the value written in this entry in SAP CPQ administration upon entering SAP CPQ through SF backdoor. If they don’t match, the user will not be able to enter SAP CPQ. If Security password for verifying entrance to SAP CPQ through the Salesforce backdoor is not empty, replace Password in the Visual Force page with the entry from SAP CPQ.

**Step 4: Override Quote’s Standard Buttons and Links**

Now that we created VF pages, we want to use them right? So we need to replace the standard pages for new, edit and view with the pages we just created. Go to Setup Build Create Objects. Scroll down to the Buttons, Links and Actions section. Here you can select the particular VisualForce (VF) page from the dropdown, which will be assigned to the desired button. To do this, click Edit button of the particular action. That leads you to the Override Properties section.

- Override View: Choose Visual Force page CPQQuoteEdit (Note: View uses the Edit page).
- Override Edit: Choose Visual Force page CPQQuoteEdit.

**Step 5: Change Opportunity Page Layout**

Go to Setup Customize Opportunities Page Layouts. You can edit all the pages for opportunity per business function. For now, click Edit for Opportunity (Sales) Layout. When you click Edit on the appropriate page layout, you will find a Quotes Section at the bottom of the screen.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>You can drag the quote section to the top so that it will display at the top of the opportunity page.</td>
</tr>
</tbody>
</table>

On Quotes area, click the “wrench” to enter properties. On this page, select the following columns and move them from Available Fields list to the Selected Fields list.

- Quote Id
- Revision Number
- Discount Percent
Step 6: Open the existing SAP CPQ quote page within Salesforce tab

To have the SAP CPQ quote page appear within Salesforce and have one, seamless interface, you must configure the following:

- Go to Setup > Build > Create > Tabs and create a New Tab by clicking the New button under the Custom Object Tabs section. You should be able to see the Quote object you created in the drop-down menu.
- Click on next twice (you do not need to change any default settings on the next two screens). When the button Save shows up, click on it to save your settings. You should now be able to see the SAP CPQ Quotes tab in the menu bar at the top of the page.
- You have now completed the SAP CPQ setup in Salesforce. Now you must set up SAP CPQ to sync with the new Salesforce object you have just created.

SAP CPQ side Setup

Log into your SAP CPQ domain and go to the Setup section. First, you must enable Salesforce integration in SAP CPQ setup, in order to get the additional menu options under CRM Integration submenu. To do so, go to Setup > CRM Integration > General. Now, go to Setup > CRM Integration Setup > CRM Quote. In the first section (Basic mappings), populate fields with the following values:

- CRM Quote Object Name: Quote__c
- CRM Field For Persisting Quote Id: Quote_Id__c
- CRM Field For Persisting Quote Owner Id: Owner_Id__c
- CRM Field For Persisting About Primary Quote: Primary__c

Click on Save before proceeding to additional mappings. Alternatively, you can click CRM Lookup on the right-hand side, and select the field. If you are within the same web browser session, this should be available. Don’t forget to click Save before proceeding to additional mappings. In the second section (additional mappings), click Add Mappings. You will need to create five mappings. Here is the list of all mappings:

- Discount_Percent__c: <ctx>(Quote.Total.AverageProductDiscountPercent.DefaultDecimal)*>
- Total_List_Price__c: <ctx>(Quote.Total.TotalListPrice.MarketDecimal)*>
- Name: <ctx>(Quote.CartCompositeNumber)*>
- Revision_Number__c: <ctx>(Quote.Revision.RevisionNumber)*>

You have now completed the Setup. Go back to Salesforce to test it out. To do so, click Opportunities in your Salesforce account, create new Opportunity, or open the existing one.
Scroll down to the Quotes section and click *New Quote*. Your quote page should open directly into Salesforce. If this doesn’t occur, go back and check your steps. Mistakes are common on Visualforce pages, especially with the URL and the domain.

**Notes**

- When test-building a quote, and watching it appear in Salesforce, the quote will appear in the Notes and Attachments section. Or, if configured properly, it will appear in the Quotes section. The products in the quote will be added to the Products Tab.
- If your products don’t have a part number, they may not transfer properly. In this case, go to Product Setup and add a product code/number to the product’s existing attributes.
- Sometimes the account and contacts in Salesforce don’t have all the proper information, so you may need to ensure your contacts have all the proper fields set up correctly (e.g. country and zip code). Also, make sure you have a functioning address.
- The *Developer Console* logs a trace of the SOAP API calls made from SAP CPQ to Salesforce. The logger label for the trace is `Webcom.Configurator.SFPartnerWSDL.SforceService`. This trace gives administrators an overview of the duration of the actions executed via the API.

**Troubleshooting**

- If you are not able to connect to SAP CPQ and it advises you to contact SAP CPQ administrator, check the VisualForce Page and ensure everything is correct, especially the domain name.
- Also check the Quote identifiers (e.g. `Quote_ID__c`). Make sure they match exactly, remember they are case sensitive and it is a double underscore, `__c`.
- If it connects to SAP CPQ but doesn’t log you in, check the user mapping and the password in `CRM integration > General`. Try removing the password or entering it.
- When you’re landing to SAP CPQ from SalesForce in Safari on iPad, you may not be able to scroll up and down the iFrame window in which SAP CPQ is opened if you’ve previously interacted with the SalesForce area until you tap on the empty screen area of the iFrame first. This also applies to the scenarios in which you’re scrolling through the SalesForce area after interacting with the iFrame. By tapping on the empty screen area of the application you’re currently working in, you’re focusing that part of the screen and the scroll works correctly.

**Related Information**

- Salesforce Package Integration Troubleshooting
- Salesforce Integration (Single Quote Method) [page 272]
9.1 Salesforce Integration (Single Quote Method)

In order to set up a link from Salesforce opportunity that points to the SAP CPQ application, administrator should perform following steps:

- Login to Salesforce and go to Setup (link is on the top of the page)
- In the menu on the left, click [App Setup ➤ Develop ➤ Pages ➤]

Controls and then click on ‘New’

- Fill the ‘Label’ and ‘Name’ (for example with the text ‘Generate Quote’)
- In the field ‘Content’ copy and paste:

```
<apex:page standardController="Opportunity" tabStyle="Opportunity">
</apex:page>
```

- Make the following 3 changes to the above URL:
  - Replace CPQInstallation with the URL for your installation, for example v60test.webcomcpq.com or v60.webcomcpq.com.
  - Replace DomainName in the above URL with the name of your SAP CPQ domain (tenant).
  - Optionally you can use apiPass. This is the security measure. In SAP CPQ admin, under CRM Integration Setup ➔ General, there is an entry ‘Security password for verifying entrance to SAP CPQ through Salesforce backdoor’. If this entry is left blank, apiPass from the Visual Force page is not verified upon entrance through backdoor. However, if this entry is not empty, upon entering SAP CPQ through SF backdoor, value written in the vf page will be matched with the value written in this entry in SAP CPQ administration. If they don’t match, user will not be able to enter SAP CPQ.

- click on Save
- in the menu on the left, click on ![App Setup ➤ Customize ➤ Opportunities ➤ Buttons and Links ➤]
- under Custom Buttons and Links click on New
- enter Label and Name (for example you could write Generate Quote - this label be displayed on Opportunity tab)
- under Behavior select Display in existing window without sidebar
- under Content Source choose VisualForce Page and find the visual force page you’ve created previously
- click on Save
- in the left menu click on ![App Setup ➤ Customize ➤ Opportunities ➤ Page Layouts ➤]
- find the page layout that is used for displaying opportunities and click on Edit.
- on the right you will see drop-down menu labeled View. Choose Opportunity Custom Links
- under this drop down the list of custom links will appear
- select the custom link you’ve previously created and drag and drop it to the area in the middle of the page labeled Custom Links (this section may also be called Useful links)

- click on Save
- Go to Opportunity tab
- Select existing opportunity or create new opportunity
- In the opportunity you’ve selected there should be custom link that you have created. Click on this link and you should be presented with SAP CPQ application.

Related Information

Salesforce Integration Instructions [page 263]
Salesforce Integration [page 255]
The General administrative section is used to define settings for the connection between SAP CPQ and the CRM.

Here a user can define how SAP CPQ will communicate with the CRM, but also he can manage some of the interaction settings. To access the General administrative section, click the link under the CRM Integration Setup admin menu.

**SAP CPQ is integrated with** - Determines which CRM application SAP CPQ will integrate with. The selection determines what CRM icon is used in the CRM integration admin sections and what is seen on the user side on the customer info tab. In addition, the options available in the General admin section change based on the CRM system used. To view a CRM’s options, select the CRM from the **SAP CPQ is integrated with** field and click Save. This displays the options specific to that CRM.

**Salesforce Edition** - Choose the Salesforce edition of the installation that SAP CPQ will be integrating with. Choose Sandbox or Standard. Default edition is standard.

**Security password for verifying entrance to SAP CPQ through Salesforce backdoor** - This field allows setting extra security measures between SAP CPQ and Salesforce integration. A user can define a password in the URL used for the custom link to access SAP CPQ. This password, set under the `apiPass` element, is checked against the password entered in this field. This acts as an extra security measure to ensure that there is no unauthorized access to SAP CPQ.

**Partner Role Type** - Used when a user searches for CRM accounts from the customer info tab. When searching, both partner and child accounts of the user’s current account are displayed. This field determines which partner accounts will be displayed based on the partner role type(s) entered here. Click **CRM Lookup** to view all partner role types on the CRM application. More than one type can be entered. Each selection made will be entered into the field separated by a comma.

**Note**

The integration is case sensitive. Use CRM Lookup to select CRM Fields.

**All revisions from the quote will be attached to the same opportunity** - Determines whether or not revisions made on an SAP CPQ quote will be attached to the same opportunity or not. This option is generally set to Yes. If the parameter is set to NO, multiple revisions of the same cart can be attached to opportunity.

**Enter CRM Account Record Type ID which will be used when new CRM Accounts are created by CPQ** - Determines the default record type for all new accounts that are created from SAP CPQSAP CPQ. Click CRM Lookup to connect to the CRM application and select the desired record type. When clicked, the name will be displayed in the field, but in the SAP CPQ database, the Record Type is being used. This feature is used mainly in the CRM integration approach of starting from SAP CPQ. Setting the record type allows you to keep track of those accounts created in the CRM from SAP CPQ.

**On customers page, users may override mappings for an account that is used for newly created opportunity** - This determines if the user is able to override which customer role will be used for the creation of a new account on the CRM. If set to Yes, then the user will be able to select the customer role on the customer info tab to be used in the opportunity. If set to No, then the default customer role will be used to create the new account in the opportunity.
Application parameter **Update existing products in Salesforce when sending data from SAP CPQ items:**

- When set to “Yes,” SAP CPQ will update the SFDC Product 2 table. This will happen even if no fields are set to be mapped to this object, and only an identifier has been put in place for the system to recognize the SF Product in SAP CPQ.
- If mapping from SAP CPQ to this object is not needed, the application parameter should always be set to “No.”

Application parameter **Quote object in CRM is NOT deleted every time action ‘Create/Update Opportunity’ is executed:**

- When set to **TRUE** system will select all Quote object records from the current opportunity, with the Owner ID and Cart ID equal to current cart in SAP CPQ and it will take the first one from this list and update it instead of creating a new Quote object. If no Quote object records existed on the opportunity with this Owner ID and Cart ID, the system will create a new Quote record.
- When set to **FALSE**, the system will delete ALL (there will usually be only one) Quote object records from SFDC Opportunity with the Owner ID and Cart ID equal to current cart in SAP CPQ for which the “create/update opportunity” is selected and will then insert a new Quote object row.

**Product types that will be/will not be included in CRM opportunity** - Defines which product types will or will not be pushed as products to the CRM. By default all product types are pushed to the CRM. This includes line items and main items. Select a product type you would NOT like to be pushed to the CRM and click the arrow.

**When editing quotes from CRM opportunity, optimize saving of customer roles** - As a way to improve performance when editing quotes from CRM opportunities, logic has been introduced that will control whether customers are saved or not depending on values that are mapped from CRM. If there are no values inserted customer roles, SAP CPQ will not save customer roles and thus save time when editing quotes from CRM. This application parameter is turned off by default and needs to be set to “TRUE” in order to turn on the optimization.

**Performance** - As a way to improve performance when users arrive from Salesforce, there is a flag that can be checked - ‘Implement Caching for Salesforce Object Definitions’. When this flag is turned on, for each user that arrives from SF, object definitions (field names, field types, permissions etc.) will be stored in SAP CPQ SAP CPQ database. This can improve performance when there is a large number of fields (more than 100) on objects opportunity, account and contact, since SAP CPQ queries Salesforce API for object definitions upon user’s arrival and it takes 2-3 seconds for SF API to return each object definition with so many fields. It’s important to know that there is ability to **Refresh cache for Salesforce object definitions**. This should be done when a new field is added in Salesforce to object definitions or when permissions have been changed over object fields in SF. If object fields or permissions have been changed in Salesforce and these fields are used in integration with SAP CPQ, errors may appear in certain integration points e.g. ‘Create/Update Opportunity’.
11 CRM Objects

SAP CPQ allows mapping of objects between SalesForce and SAP CPQ – if users arrive from opportunity to SAP CPQ and create a quote, the opportunity custom fields can be copied to appropriate SAP CPQ custom fields. Also opportunity account, partner or contact roles can be mapped to SAP CPQ customer roles, etc.

In this part of SAP CPQ Setup, the administrator can define Salesforce objects that will be mapped to SAP CPQ objects.

SAP CPQ supports several integration objects as part of the “out-of-the-box” solution. These objects are not custom (flag Is Custom is set to No) – objects like Opportunity, Opportunity Account, Opportunity Primary Role, Opportunity Bill To Role etc. The administrator cannot edit these objects.

To create an integration object go to Setup > CRM Integration Setup > CRM Objects and click Add New. The following fields are available:

1. **Name** of the object. You will use this name to identify the object when setting up custom fields or customer roles mappings.
2. **Type** of object you wish to create (Account, Contact, etc.). Use CRM Lookup to select type.
3. **Select whether users will be able to attach this object to the quote**. For most objects this checkbox will be unchecked. If this checkbox is checked, users will be able to link this object with a SAP CPQ quote. Opportunity is an object that can be linked to quotes by default, and the action Create/Update Opportunity is used

   - If **Select whether users will be able to attach this object to the quote** is checked, choose the action that will create this link (for now, only the action Create/Update Custom Object is available for this functionality). It is during this action that SAP CPQ will persist the connection between quote and selected Salesforce object in SAP CPQ database.
   - If **Select whether users will be able to attach this object to the quote** is unchecked, enter a Query that will be used for querying the SalesForce API for this object. Use the Formula Builder for expressions with tags.

**Examples**

Task: Get the opportunity partner role account ‘Supplier’

Name: Opportunity Supplier

Type: Account

Select whether users will be able to attach this object to the quote: No

Query: SELECT AccountTolId FROM Partner WHERE OpportunityId =CTX(SFDC.Opportunity.Id) AND Role = 'Supplier'

Task: Get the opportunity contact role account ‘Decision Maker’

Name: Opportunity Decision Maker

Type: Contact
Select whether users will be able to attach this object to the quote: No

Query: `SELECT ContactId FROM OpportunityContactRole WHERE OpportunityId = '<*CTX(SFDC.Opportunity.Id)>' AND Role = 'Decision Maker'`

**Note**
SAP CPQ supports only simple queries made to SalesForce API. Administrators cannot use aliases in queries.

**Note**
If custom objects are to be used in Quote Create/Update event, the above examples are valid. If these objects were to be used in Create/Update Opportunity or in Create/Update Custom Object action, instead of `<*CTX(SFDC.Opportunity.Id)>` tag `<*CTX(Quote.Opportunity.Id)>` should be used.
The **Custom Fields Mapping** administrative section allows you to map fields, custom or standard, from SAP CPQ to opportunity or opportunity account fields in the CRM.

To access this administrative section, click the link under the **CRM Integration Setup** admin menu.

Mappings can be set for either of the two integration approach methods: **Starting from CPQ** and **Starting from the CRM**. The tabs at the top of the **Custom Fields Mapping** administration page allow you to define mappings for either integration method. From here you can **Copy**, **Delete**, or **Edit** existing mappings.

To define a new mapping, click **Add New**.

At the top of the definition screen choose the Event that will initiate this mapping. The following events are available:

**Starting from the CRM:**

- **Quote Create** is initiated when the user clicks Add to Quotation to create the quote.
- **Quote Update** is initiated when the user enters the quote from the opportunity.
- **Quote Create Update** is initiated on both the create event and the update event. Starting from SAP CPQ:
  - **Opportunity Create** is initiated when the Create/Update Opportunity action is clicked. If this action has been set as a pre or post action of another action, then the event will run when that action is clicked. The values will only be passed from or to the CRM on opportunity creation, and not when an opportunity is updated.
  - **Opportunity Update** is initiated when the Create/Update Opportunity action is clicked. If this action has been set as a pre or post action of another action, then the event will run when that action is clicked. The values will only be passed from or to the CRM on an update, and not on opportunity creation.
  - **Opportunity Create Update** is initiated when the Create/Update Opportunity action is clicked. If this action has been set as a pre or post action of another action, then the event will run when that action is clicked. Values will be passed from or to the CRM when the opportunity is created and updated.

In the **CPQ Fields** section, select the type of field that will be mapped. Choosing Custom Quote Fields allows you to choose any custom field created in the system. Choosing Stand Quote Fields allows you to select the following values:

- **Quote Number** returns the quote number that will be associated with the opportunity
- **Total Amount** returns the total amount of the quote
- **Quote Description** returns the description of the quote.

In the **CRM Fields** section, select which field type the SAP CPQ field will be mapped to: **Opportunity** or **Opportunity Account**. Next, click the **CRM Lookup** link to select the particular CRM field that you would like mapped. This displays a pop up window where you can select the desired field.

---

**i Note**

The integration is case sensitive. Use CRM Lookup to select CRM Fields.
The CRM Lookup link uses the CRM administrator account defined in the CRM Administrator Account administrative section to connect to the CRM. You must have this account defined for the CRM Lookup link to work properly.

Once you have the fields defined that will be involved in the mapping, click the arrows in the Direction column to determine which system will drive the value of the field mapping. The up arrow designates SAP CPQ values to be passed in this mapping. The down arrow designates the CRM values to be passed in this mapping. The system that drives the values in the mapping will overwrite any values in the other system. For instance, if you have set up a mapping in which SAP CPQ drives the values in a mapping with the CRM, then the values on the CRM will be overwritten by the SAP CPQ values when the mapping is initiated.

When Standard SAP CPQ fields are being used in the mapping, the only direction available is arrow pointing to the right. This is because the standard fields, such as Total Amount and Quote Number, are defined by the user’s configuration and cannot be changed by another system.
13 Customer Roles Mapping

The Customer Roles Mapping administrative sections allows you to set up the mappings between SAP CPQ customer roles, such as Bill To, Ship To, End User, and the opportunity information.

To access this administrative section, click the link under the CRM Integration Setup admin menu.

Mappings can be set for either of the two integration approach methods: Starting from CPQ and Starting from the CRM. The tabs at the top of the Customer Roles Mapping administration page allow you to define mappings for either integration method. Because only one mapping can be defined per integration approach, editing the mapping is just a matter of changing the fields and saving the new settings.

Defining mappings for customer roles is done in a two step process. The first step is defining the SAP CPQ customer role, then selecting the CRM object that the role will be mapped to. The second step is to define the field level mappings where you define the field from the CRM account to be mapped to the SAP CPQ customer role field.

First, choose the SAP CPQ customer role that the mapping will be defined for. The arrow between the SAP CPQ roles and the CRM icon represents which system will drive the values. When the integration approach method is starting from the CRM, then the CRM will fill the customer role information in SAP CPQ. Alternatively, when the integration approach method is starting from SAP CPQ, then SAP CPQ will fill the account information in the CRM.

If you have already defined object level mappings for one for the customer roles, you can easily copy the same settings to another role. In the right corner of the page is the Copy settings from box where you can choose the role to copy the settings from. This will only copy object level mappings to this role, and will not copy any field level mappings.

The Account and Contact selections determine from which account and contact will be involved in the mapping. Depending on the CRM being used, the choices available may be different. In the event that this choice is not available when the mapping occurs, you can set alternative accounts and contacts to be used in the mapping. This is done through the Additional Options panel, which is described in more depth later.

The Event selection determines when the mapping will be initiated. The following events are available:

Starting from the CRM:
- **Quote Create** is initiated when the user clicks Add to Quotation to create the quote.
- **Quote Update** is initiated when the user enters the quote from the opportunity.
- **Quote Create Update** is initiated on both the create event and the update event.

Starting from SAP CPQ:
- **Opportunity Create** is initiated when the Create/Update Opportunity action is clicked. If this action has been set as a pre or post action of another action, then the event will run when that action is clicked. The values will only be passed from or to the CRM on opportunity creation, and not when an opportunity is updated.
- **Opportunity Update** is initiated when the Create/Update Opportunity action is clicked. If this action has been set as a pre or post action of another action, then the event will run when that action is clicked. The values will only be passed from or to the CRM on an update, and not on opportunity creation.
Opportunity Create Update is initiated when the Create/Update Opportunity action is clicked. If this action has been set as a pre or post action of another action, then the event will run when that action is clicked. Values will be passed from or to the CRM when the opportunity is created and updated.

The Additional Options panel defines alternative mappings in the event that the first choice of account and contact is not available when the mapping occurs. Clicking the Additional Options bars displays the panel.

To define alternative account and contact that should be used, select the account or contact from the list and click the down arrow to bring the item into the selected area. Likewise, any selected item can be unselected by clicking the up arrow. You can further prioritize selected item by clicking the up and down arrows.

**i Note**
The Additional Options pane is not available for the Starting from SAP CPQ integration approach. This is because you are not defining where the values are coming from, but instead what account and contact values SAP CPQ should fill on the CRM.

Once you have selected the account, contact, and SAP CPQ customer role that will be involved in the mapping, click the Continue to Field Level Mappings button. This screen allows you to define mappings between the SAP CPQ field and the CRM field.

If you have already defined field level mappings for one for the customer roles, you can easily copy the same settings to another role. In the right corner of the page is the Copy settings from box where you can choose the role to copy the settings from. This will only copy field level mappings to this role, and will not copy any object level mappings.

For each SAP CPQ field, you can define the CRM field type, Account or Contact, then define the specific field that should be mapped. Click the CRM Lookup link to select the specific CRM field. This displays a popup window where you can select the CRM field.

**i Note**
The integration is case sensitive. Use CRM Lookup to select CRM Fields.

**i Note**
The CRM Lookup link uses the CRM administrator account defined in the CRM Administrator Account administrative section to connect to the CRM. You must have this account defined for the CRM Lookup link to work properly.

Not every field is required for integration. You only need to define mappings for those necessary fields. Keep in mind, though, that any SAP CPQ customer role fields set as required (defined in Customer Field Permissions admin section) will require the user’s input if not filled from the CRM account/contact values.
14 Opportunity Status

This section defines the mappings between a CRM opportunity status and a SAP CPQ quote status. These mappings are only done when the Create/Update Opportunity action is initiated, either by the user clicking the action or by a pre or post action.

To create a new mapping, click Add New. On the opportunity mapping definition page, choose the CPQ Cart Status from the list box. To choose the CRM Opportunity Status, click CRM Lookup. This displays all the opportunity statuses from the CRM application.

i Note

The integration is case sensitive. Use CRM Lookup to select CRM Fields.

When a status is selected, it is automatically entered into the CRM Opportunity Status field.
15 CRM Item Mappings

CRM Item Mappings is used to transfer the Quote item level information from SAP CPQ to SFDC when the Create/Update Opportunity action is executed.

To access this administrative section, click [CRM Integration Setup] ➤ CRM Item Mappings.

Two tabs are available on the CRM Integration Setup page: Action 'Create/Update Opportunity' and Action 'Create/Update Objects From Quote Items'.

Action 'Create/Update Opportunity'

This tab allows admins to define how their quote items are sent to standard SFDC objects. It defines which quote item level fields will be sent to SFDC when executing the “Create/Update Opportunity” action. In this tab, admins can only write into Product2 and OpportunityLineItem objects (standard SFDC objects).

Action 'Create/Update Objects From Quote Items'

This tab allows admins to write from CPQ Quote Item level fields to non-standard SFDC objects. Admins first define the non-standard SFDC object (in the Define Objects section), and then define the mappings between the custom object and Quote items (in Define Mappings section below).

To define a new mapping:

1. Click Add New to open the Custom Item Mapping definition screen.
2. In Standard Quote Fields enter the SAP CPQ field that will be mapped to the CRM field. Use the Formula Builder to enter expressions with tags.
3. Select the CRM item field type - Product or Opportunity Line Item.
4. Click CRM lookup to select the field.
5. Select Identifies Product to use the SAP CPQ field to identify a product in CRM.
6. Click Save to finish.

To edit or delete a mapping, find the mapping under [CRM Integration Setup] ➤ CRM Item Mappings and click the Edit or Delete icon respectively.

Examples

Standard Quote Fields: <CTX(Quote.CurrentItem.PartNumber)>

CRM item field type: Product CRM field: Name
Identifies Product: True

The value of the SAP CPQ field, in this mapping the Item Part Number, will be used to identify the Product in CRM.
16 Price Book Market Mappings

This section determines what SAP CPQ market to apply to products coming from a CRM pricebook. This helps to determine what currency is being used and how to calculate the price of the product. In addition, when an opportunity is created the product included in the shopping cart is automatically created, if it doesn’t already exist, and placed into the correct price book on the CRM application.

To create a new mapping, click Add New. On the market price book mapping definition page, select the SAP CPQ market from the Market Name field. Select the CRM Price Book ID by clicking CRM Lookup and selecting the desired Price Book.

i Note
The integration is case sensitive. Use CRM Lookup to select CRM Fields.

Once the price book mappings are defined, the correct price book will be chosen during an opportunity creation and update. There are four different scenarios possible during price book mappings:

Starting in CRM and creating a new quote in CPQ | The price book chosen for the opportunity will automatically select the SAP CPQ market based on the mappings.

Starting in CRM and opening an existing quote in CPQ | The price book chosen for the opportunity will automatically select the SAP CPQ market based on the mappings. This could potentially change the SAP CPQ market if the opportunity’s price book has changed since the last time the Create/Update Opportunity action was performed.

Starting in CPQ and creating a new opportunity in CRM | The price book is chosen based on the mappings with the selected SAP CPQ market for the quote.

Starting in CPQ and updating an existing opportunity in CRM | The price book is chosen based on the mappings with the selected SAP CPQ market for the quote. This could potentially change the price book if it had been changed by a user outside of a SAP CPQ Create/Update Opportunity action.

i Note
An opportunities price book will be changed if a user comes from the CRM to an existing quote, changes the market (which has been selected based on the price book), then performs the Create/Update Opportunity action.
17 User Mappings

The User Mappings administrative section allows you to control the mappings between accounts on the CRM and accounts in SAP CPQ. This section shows the mappings that have been created already between the CRM and SAP CPQ.

The first time a user logs into the CRM and accesses SAP CPQ through the custom link created in the opportunity, they are asked to log into SAP CPQ. Once logged into SAP CPQ, a mapping is created between their CRM account and the SAP CPQ account they logged in as. In the future when they access SAP CPQ through the custom link in the opportunity, they are not asked to log into SAP CPQ. This mapping created between the two user accounts is managed in the User Mappings administrative section. So make sure that users are actually able to log in for the first time. (Don’t set them as inactive in administration). If they ever become inactive, they won’t be able to access SAP CPQ from CRM any more.

The User Mapping admin section can be accessed by clicking the link under the CRM Integration Setup admin menu. The section lists every user in the system. Users with mappings already defined display the mapped CRM user in the second column. Other users without a mapping show nothing in this column.

To delete a mapping between a SAP CPQ user account and a CRM user account, click the delete button. This breaks the mapping and forces that CRM user to log into SAP CPQ the next time they access SAP CPQ through the custom link created in the opportunity.

To edit a mapping or add a mapping between two user accounts, click the edit button. The SAP CPQ user you’ve edited will be the only user available to edit. The field will not allow you to edit the SAP CPQ user. Likewise, if you are editing an existing mapping, you will be unable to modify the CRM user involved in the mapping.

To change the CRM user involved in the mapping, click the CRM Lookup link. This displays the search results of what is entered in the search field. Leave the search field empty to return all the users in the search results. Select the CRM user you would like mapped with the SAP CPQ user. Once the mapping is defined, the user will not be required to log in to SAP CPQ after clicking the custom link in the opportunity.

⚠️ Caution

CRM users can be used in only one mapping with a SAP CPQ user. The system will not save the mapping if the CRM user is being used with another SAP CPQ user.

Also, before you rename user in CRM, a mapping in SAP CPQ should be removed, and then user remapped.
The **CRM Quote** administrative section allows mapping SAP CPQ tags to fields, custom or standard, from the **Salesforce Quote** object. Access to administrative section is possible by clicking the link under the **CRM Integration Setup** admin menu.

**i Note**

To see how to set up multiple links from Salesforce opportunity to SAP CPQ quotes, first read the chapter **Salesforce Integration (Single Quote Method)** [page 272].

For instructions on how to set up SAP CPQ Quotes for CRM OnDemand refer to **Setting up Quote object and links in CRM On Demand**.

The **CRM Quote** administrative section allows mapping SAP CPQ tags to fields, custom or standard, from the **Salesforce Quote** object. Access to administrative section is possible by clicking the link under the **CRM Integration Setup** admin menu.

In the first section, the administrator can define:

- **CRM Quote Object Name** – API name of the custom created ‘Quote’ object in Salesforce.
- **CRM Field For Persisting Quote Id** – API name of the field in the ‘Quote’ object used for storing information about the quote Id.
- **CRM Field For Persisting Quote Owner Id** – API name of the field in the ‘Quote’ object used for storing information about the quote owner Id.
- **CRM Field For Persisting Information About Primary Quote** – API name of the field in the ‘Quote’ object used for storing information whether the quote is primary or not for this opportunity.
- **CRM Field For Persisting Information About Quote Currency** – API name of the field in the ‘Quote’ object used for storing information about the quote currency.

**i Note**

The integration is case sensitive. Use CRM Lookup to select CRM Fields.

Click on Save before proceeding to additional mappings.

In the second section, the administrator can define other SAP CPQ data that will be pushed to the Salesforce ‘Quote’ object. That can be Total List Price, Total Net Price of the quote etc. The administrator needs to define the field in the Salesforce ‘Quote’ object, used for storing the value, and the SAP CPQ value – by using standard SAP CPQ tags.

By clicking **Add New Mapping**, the administrator can add a new mapping – ‘Quote’ object field and SAP CPQ tag.

---

**SAP CPQ Integration Guide**

**CRM Quote**

PUBLIC 287
19 CRM Administrator Account

This section is used for defining the administrator username and password used to access the CRM Application. The account is used to access the CRM application for lookups in administrative sections.

In addition, it is used in the integration approach method of starting from SAP CPQ. The account set here as the administrator account will be used to connect to the CRM to create and update opportunities and accounts/contacts.

You can define multiple administrator accounts if desired. This can be useful if you have multiple installations of the CRM application, such as a development installation and a production installation. You can then switch between the installation that SAP CPQ integrates with by setting the administrative account of the installation you would like to use as the active account. Keep in mind, though, that doing this will only work if the same fields, opportunity status, and other mappings are the same in the development and production installations.

1. Go to Setup > CRM Integration > CRM Administrator Account.
   The page with all CRM administrators in SAP CPQ is listed.
2. Click Add New to define a new CRM administrator.
   The account definition page displays.
3. Select the type of CRM that the account refers to.
4. Enter a username and a password.
5. (Optional) Select the Is Administrative Account to set this account as the active administrator account.

   ![Note]
   All passwords are encrypted in the database for security reasons.

Salesforce Security Token

Due to Salesforce security measures, a security token must be added to the password. If you do not know what your security token is, you can obtain it from Salesforce by logging into SFDC and going to Setup > Personal Setup > My Personal Information > Reset My Security Token. If you activate this action, a new security token will be sent to your email address. You should then enter (in SAP CPQ administration) your Salesforce username, and in the password field, you should enter your password and security token joint together (if password = “mypassword” and security token = “XXXXXXXXXX”, you should enter “mypasswordXXXXXXXXXX”).

Landing from Salesforce to SAP CPQ

When you are landing to SAP CPQ for the first time from Salesforce, a new user is automatically created for you in SAP CPQ. However, if a regular user wishes to land to SAP CPQ for the first time, it is necessary for their...
SFDC system administrator to have a CRM administrator account in SAP CPQ. Therefore, *Is Administrative Account* must be set to *True* on the *CRM Administrator Account* page.
20  Send Document To CRM

It's possible to send documents to CRM as pre or post-action to another action.

All you have to do is to set *Send Document To CRM* action as pre or post action to desired action (e.g. *Generate Quote*) for adequate status in *Workflow*.

This action is basically intended to be used as post-action to *Generate Quote* for Salesforce integration.

SAP CPQ will show error message if error happened in communication with CRM during document sending.

If SAP CPQ is sending file to Salesforce and if file size is too large for Salesforce.com to accept it, SAP CPQ will not attempt to send it automatically and error message will be displayed to user.

After successfully sending the document to Salesforce, it will appear on Opportunity in *Notes & Attachments* section.

---

**i Note**

To make action *Send Document to CRM* visible on the *Download Document and Send Email* page, set application parameter *Hide option to send document to opportunity from user* (located in the *Email Customers/Quote Acceptance Settings*, in the *Quotes* submenu) to *False*.
21 New Authentication method for COD integration

The purpose of this feature is to improve security for COD-SAP CPQ integration. SAP CPQ will no longer store COD username and password.

User Side

The change can be applied to all links created from COD to SAP CPQ (Generate Quote links, Quote object links etc.). If user hasn’t been already mapped, screen for entering log-in info will be provided to user. Screen will be different depending on the value of application parameter Prompt Users to Enter COD Username/Password when coming first time to CPQ - if value is TRUE, page will remain the same.

Setup

In SAP CPQ SetupCRM Integration SetupGeneral, there will be new application parameter that will control whether users need to enter COD un/pass. Label for parameter will be Prompt Users to Enter COD Username/Password when coming first time to CPQ (default value will be TRUE).
Netsuite Integration With SAP CPQ

To integrate Netsuite with SAP CPQ, follow next steps.

**Step 1: Log in**

- Go to [http://www.netsuite.com](http://www.netsuite.com) and enter your credentials.

**Step 2: Create a new script**

- Download `CPQSSOSuitelet.js` JavaScript file from the following link (.js file is zipped): [CPQ SSO Suitelet](#).
- Unzip the file on your computer.
- At the top menu go to `Customization > Scripting > Scripts > New`.
- On the next screen, click on "plus" button. Note that this button appears if you hover your mouse pointer over the text field.
- In the pop up window, fill in text fields as follows:
  - Attach from: `Computer`
  - File Name: `CPQSSOSuitelet.js`
  - Folder: `SuiteApps`
- Pick the `CPQSSOSuitelet.js` file you saved on your computer in previous step
- Click `Save` button. This will close the pop up
- On the next screen click on `Create Script Record` button
- Pick a script type on the next screen
- Enter the the name in the `Name` text-entry field. You don’t need to populate the field ID because it will be populated by default later.
- Populate the `Function` field in the `Scripts` tab with the following: `buildSuitelet`.
- Click `Save`. Script details screen is displayed.

**Step 3: Deploy script**

- Click `Deploy Script`. `Script deployment` page is displayed.
- Populate the following fields:
  - Status - `Released`
  - Log level - `Error`
- Click `Save`. Details screen is displayed
Step 4: Edit Script File

- Go to [Customizations > Scripting > Scripts]. Find your script and click on view link. Script details page is displayed.
- Click on edit beside script file name.
- In CPQSSOSuitelet.js file update the value for the variable suiteletId with the value you wrote down in Step 3.
- In step 6, you will create SSO integration. Once you create it, you should update this line of code in CPQSSOSuitelet.js file:
  ```javascript
  var url = nlapiOutboundSSO('customsso1');
  ```
  with this code:
  ```javascript
  var url = nlapiOutboundSSO('YourIntegrationID');
  ```

Note

YourIntegrationID is the ID assigned to SSO Integration you will create in step 6

Step 5: Create opportunity form

- Go to [Transaction > Sales > Create Opportunities].
- On the same page, at the top right corner click on [Customize > Customize Form link].
- Under the Tabs tab, populate the form.
- Download OpportunityInit.js file from the following link (.js file iz zipped): Opportunity Init.
- Unzip file on your computer.
- Click the Custom Code tab, and upload script file OpportunityInit.js.
- Type OpportunityInit.js as file name.
- When file is uploaded, click the Open icon on the right-hand side.
- At the screen that opens, click Edit for OpportunityInit.js.
- Change functions NewQuote and ViewQuote so that the number located in their variables URL, following the argument script, is the same number you wrote in Step 3. Click Save button.
- Return to the Custom Code tab and, in the Page Init Function field enter initFunction.
- Go to [Action > Custom Actions] to add two custom actions: New CPQ Quote and View CPQ Quote. Functions that will be mapped to these two actions are defined in OpportunityInit.js that you uploaded earlier.
- Click **Add** to add row
- Enter **LABEL** text, **FUNCTION** name, and **DISPLAY AS**
- Once you complete steps for both functions, click **Save**

**Step 6: Create integration with SAP CPQ**

- Go to **Setup** &gt; **Integration** &gt; **SuiteSignOn** &gt; **New** &gt; Once the page opens, fill the form out with the appropriate data. Note that a Shared Secret key is defined by you, and you use it on SAP CPQ side as well.
  - Under **Connection Points** tab, you define SAP CPQ integration URL and integration variables. Enter your SAP CPQ domain name in the form `domain=Your_CPQ_domain_name`.
- **ID** that is assigned to your newly created Single Sign On must be inserted into CPQSSOSuitelet.js file.
- Under **User Identification** tab, tick two checkboxes - **account** and **email** and click on **Save** button.

⚠️ **Caution**

Write down your Consumer Key and Shared Secret because you will need them later.

**Step 7: Integration setup on SAP CPQ side**

- Go to **SAP CPQ** &gt; **Setup** &gt; **CRM Integration** &gt; **General**
- Choose **NetSuite** from the **SAP CPQ is integrated with** drop-down menu
- Fill out **NetSuite SingleSignOn url** field with the appropriate value.

*Note* NetSuite Single SignOn URL for the NetSuite development environment:

`https://system.na1.netsuite.com/app/common/integration/ssoapplistener.nl`

NetSuite Single SignOn URL for the NetSuite production environment:

`https://system.netsuite.com/app/common/integration/ssoapplistener.nl`

*Note* NetSuite Single SignOn URL for the NetSuite development environment:

`https://system.na1.netsuite.com/app/common/integration/ssoapplistener.nl`

NetSuite Single SignOn URL for the NetSuite production environment:

`https://system.netsuite.com/app/common/integration/ssoapplistener.nl`

- Fill out **NetSuite Consumer Key** and **NetSuite shared Secret** fields with the values you noted in the previous step.

Now you will have to add an administrative account. To do so, follow these steps.
Go to Setup > CRM Integration > CRM Administrator Account.

Click Add New to create a new administrator account.

Choose NetSuite as the CRM Name, then fill out other fields based on your data and needs and click Save.

The next step is to create a custom action in order to update NetSuite opportunity with SAP CPQ data.

To create new Custom Action, follow next steps.

- Go to Setup > Develop > Custom Actions, then click on Add New button.
- Fill out fields (Action Name, Placement, Scripting Code, Script).
- Download updateNetsuite.py Script from the following link (.py file is zipped): Update NetSuite
- Unzip the file on your computer, copy script text and paste Custom Action Script text field.

Set Workflow Permissions and Workflow Actions for this Action on the appropriate Setup page. For more information on how to set this, visit appropriate help pages:

- Workflow
- Workflow Permissions

**Customer Mappings**

Now, you should write a script for Customer Mappings or you can use the script attached below. To define Customer Mappings Script:

- go to Setup > Develop > Global Scripts and add a new Script.
- On tab Script give script a name (for example CustomerMappings).
- Download Script from the following link (.py file is zipped): Customer Mappings
- Unzip file on your computer, copy script text and fill out the appropriate text field.
- On tab Event, mark that the Script should execute on onCustomerMappings event.

⚠️ Caution

Note that Integration is 100% customizable – you must write your own scripts based on business needs or use scripts listed on this page, otherwise, Integration will not work.

**Custom Field Mappings**

Now, you should write a script for Custom Field Mappings or you can use the script attached below.

- go to Setup > Develop > Global Scripts and click on Add New button.
- On Script tab, define a script name (e.g. CustomFieldsMappings).
- Download Script from the following link (.py file is zipped): Custom Field Mappings
- Unzip file on your computer, copy script text and paste into the script text field.
- On Events tab, define that this script should execute on arriving from CRM and On Quote Fields Mappings event. Click Save.
Each of these scripts must be attached to an adequate Arrive From CRM event. Additional mappings (opportunity statuses, countries...) should be defined in SAP CPQ custom tables. Examples of defining such custom tables can be found in excel files, “NSTOCPCOUNTRYMAPPING” and “OpportunityStatusMapping”. Examples of using them can be found in “UpdateNetSuite” and “CustomersMapping” scripts.

Sample custom tables for country and opportunity status mappings can be downloaded from the following locations:

- NetSuite to SAP CPQ Country Mapping
- Opportunity Status Mapping

Send document to NetSuite

In order to be able to send document SAP CPQ to NetSuite, you should write an Iron Python script and attach it to the custom action that you previously defined as well. Follow next step do implement this feature.

- Go to Setup Develop Custom Actions and click on Add New button.
- Fill out the fields with the appropriate data.
- Download Custom Iron Python script from this link (.py file is zipped): Send Doc to NetSuite. Unzip the file, copy script text and paste into the appropriate field.
- Save custom action

Now, you should set availability and permission for this custom action through Workflow. To find more details on how to set Workflow, visit Workflow and Workflow Permissions help page.

User mappings

NetSuite and SAP CPQ users are mapped the first time you arrive from NetSuite to SAP CPQ (while opening or creating the quote). Only the first time that you arrive from NetSuite, you will be prompted to enter your SAP CPQ username, password and domain name. From that moment, your NetSuite and SAP CPQ users are mapped, and you do not need to enter data again.

That being said, there is no need to do manual mappings neither through SAP CPQ setup nor NetSuite setup.

Related Information

NetSuite Integration Troubleshooting
Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information. About the icons:

- Links with the icon 🌐: You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP’s gross negligence or willful misconduct.

- Links with the icon 🔗: You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP’s gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.