PUBLIC
2026-10-22

SAP Service Cloud Version 2 Utilities Integration with SAP S/4HANA

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

Content

2 Utilities Integration with SAP S/4HANA. 4

3 Prerequisites. 5

4 Configuration in SAP Service Cloud Version 2. 6

4.1 Create Communication System. 6

4.2 Create Communication Configurations. 7

5 Configuration in SAP S/4HANA. 9

5.1 Maintain Trust between SAP Cloud Integration and SAP S/4HANA. 9

Upload SAP Cloud Integration Certificate to S/4HANA. 10

5.2 SAP S/4HANA Web Service Configuration (SOA Management). 11

6 Configuration in SAP Cloud Integration. 15

6.1 Copy Integration Package. 15

6.2 Deploy Security Credentials. 16

6.3 Configure and Deploy Integration Flows. 16

Configure and Deploy Each Integration Flow Individually. 16

Configure and Deploy all Integration Flows Concurrently. 18

6.4 Configure Master Data Replication. 19

7 Configuration in SAP Cloud Connector 21

8 Replication of Master Data Initial Load. 23

8.1 Setting Up SAP S/4HANA for Contract Account Replication. 23

Creating the Outbound Service for Contract Accounts in the SOA Manager Tool. 23

Configuring the Data Replication Framework for Contract Account Replication. 24

Configuring the Delta Replication of Contract Accounts. 25

8.2 (Deprecated) Technical Master Data Replication. 26

8.3 Technical Master Data Replication. 28

Integration in SAP Service Cloud Version 2 for Utilities. 28

Integration in SAP S/4HANA. 28

Integration in CPI. 29

8.4 Data Replication for Point of Delivery. 30

Setting Up SAP S/4HANA Utilities for POD Replication. 30

Integration in SAP S/4HANA. 31

Maintain IMG Documentation. 32

8.5 Load Business Partner and Business Partner Relationships. 33

9 Principal Propagation. 34

9.1 Configure Trust. . . . 35
9.2 Configure Principal Propagation. . . . 36

10 Extensibility. . . . 38
10.1 Create Extension Fields. . . . 38
Extension Field Data Types. . . . 41
Field Attributes. . . . 43
10.2 Create Extension Fields in SAP S/4HANA for Utilities. . . . 45
10.3 Integrate Using Cloud Integration (CI). . . . 46
Custom Extensions. . . . 46

11 Appendix. . . . 51

12 Utilities Sales Process Integration with SAP S/4HANA. . . . 52
12.1 Prerequisites. . . . 52
12.2 Configuration in SAP Service Cloud Version 2. . . . 53
Create Communication System. . . . 53
Create Communication Configuration. . . . 54
12.3 Configuration in SAP S/4HANA. . . . 55
Prerequisites. . . . 55
Maintain Trust between SAP Cloud Integration and SAP S/4HANA. . . . 55
SAP S/4HANA Web Service Configuration (SOA Management). . . . 56
12.4 Configuration in SAP Cloud Integration. . . . 58
Copy Integration Package. . . . 58
Deploy Security Credentials. . . . 59
Configure and Deploy Utilities Sales Process Integration Flows Individually. . . . 59
Configure Utilities Sales Master Data Replication. . . . 61
12.5 Configuration in SAP Cloud Connector. . . . 62
12.6 Replication of Master Data Initial Load. . . . 63
Setting Up SAP S/4HANA for Contract Account Replication. . . . 63
Technical Master Data Replication. . . . 63
Data Replication for Point of Delivery. . . . 64
Load Business Partner and Business Partner Relationships. . . . 67
12.7 Principal Propagation. . . . 67
Configure Trust. . . . 68
Configure Principal Propagation. . . . 68

2 Utilities Integration with SAP S/4HANA

This document describes the procedure to manually configure the integration between SAP S/4HANA and SAP Service Cloud Version 2 for Utilities Add-On using SAP Cloud Integration (SCI).

SAP does not deliver the configuration steps as they are customer specific, and those must be completed by the customer. This document describes the general configuration steps to manually set up the configuration within the existing system landscape. If you have any queries or feedback about this document, please create a ticket using the component *CA-GTF-INT-CFC*.

If you have any queries or feedback about the Integration Package SAP Service Cloud V2, Add-On for Utilities Integration with SAP S/4HANA mentioned in this document, please create a ticket using the component *CEC-CRM-UTIL*.

Related Information

Utilities

3 Prerequisites

Prerequisites for integrating SAP S/4HANA with SAP Service Cloud Version 2.

You've access to SAP Service Cloud Version 2 system which includes:

- Initial user and access information for your SAP Service Cloud Version 2 tenant provided via email by Cloud Management Service.

You've access to SAP Cloud Integration system. This includes:

- Integration developer persona assigned to your user in your SAP Cloud Integration tenant to import and deploy integration flows on SAP Cloud Integration.
- Initial user and access information for your SAP Cloud Integration tenant.
- SAP Cloud Integration Client Certificate.

Required Information

While performing the activities described here, you must enter or provide system-specific information. Before you start the integration process, it's recommended that you have the information listed. The information ensures a smooth and efficient integration of the SAP S/4HANA and SAP Service Cloud Version 2.

- SAP Service Cloud Version 2
- SAP S/4HANA system
- SAP Cloud Integration
- SAP Cloud Connector

4 Configuration in SAP Service Cloud Version 2

This integration package enables you to integrate master data between SAP Sales Cloud and SAP Service Cloud Version 2 and SAP S/4HANA.

Configuration in SAP Service Cloud Version 2 in the Core Integration Guide (Set-up Instructions for SAP Service Cloud Version 2, SAP Cloud Integration, and SAP S/4HANA) briefly explains the procedure to configure.

For more information refer to SAP Sales Cloud and SAP Service Cloud Version 2 Integration for Master Data with SAP S/4HANA or SAP S/4HANA Cloud

4.1 Create Communication System

A communication system represents an external system that is used for application integration. Communication systems can be, for example, an external time recording or master data system.

Context

On the *Communication Systems* page, you can filter the communication systems by status and authentication type. You can create and edit communication systems to exchange data across systems electronically.

Procedure

1. Log in to the system as an administrator.
2. Navigate to user profile on the right-top corner, and access the *System Settings* page.
3. Go to *All Settings* > *Integration* > *Communication Systems*.
4. Click the + (Add) icon on the *Communication Systems* page, enter the *Display ID* and *Description* for the communication system.

   > **Note**
   >
   > Display ID must be the Business System ID of your external system.

5. Click *+ Create* button in the *Inbound* tab, enter a password or upload a certificate file for authentication.
6. Click *Save and Activate* to create the inbound communication system.

> **Note**
>
> By default the user is created while creating communication system. The communication user ID is same name as the communication system and the password is maintained in the inbound communication tab of the communication system.

7. Click *+ Create* button in the *Outbound* tab, choose a protocol, enter a host name, select an authentication method from the dropdown, and click *Save and Activate* to create the outbound communication system.

Results

You've now successfully created a communication system, which you can use for integrating applications.

Related Information

Create Communication Systems

4.2 Create Communication Configurations

Configure communication between SAP Service Cloud Version 2 and the SAP S/4HANA you are integrating it with.

Context

Procedure

1. Go to *User Menu* > *System Settings* > *All Settings* > *Integration* > *Communication Configuration* .
2. To integrate Service Cloud Utilities Master Data with SAP S/4HANA, select the following communication configurations:
   - Integrate Service Cloud Financial Customer Care with SAP S/4HANA - Collections
   - Integrate Service Cloud Financial Customer Care with SAP S/4HANA - Contract Account
   - Integrate Service Cloud Financial Customer Care with SAP S/4HANA - Common
   - Integrate Service Cloud Financial Customer Care with SAP S/4HANA - Financials

   You can also click + (Create) to create a communication configuration for your requirements.

3. Click (Copy).
4. Go to the *Communication Configuration* page and click the copied communication configuration.
5. Edit *Communication System* and select the name of the communication system.
6. Repeat the same for all the communication configurations which you copied.

   Target message entity is the structure in which the target system expects the data to arrive. *Target Message Entity API Path* is the path to which the data is stored. Source entity is the structure in which the source sends the data to connector.

   The test icon gives the connection status.

   > **Note**
   >
   > - The *Target Message Entity API Path* must match the respective iflow endpoints.
   > - For all the modules except Integrate Service Cloud Utilities Master Data with SAP S/4HANA, the synchronous calls and services list are under HTTP data. For Integrate Service Cloud Utilities Master Data with SAP S/4HANA, the synchronous calls and services list are under Inbound Configuration.

7. In *HTTP Data*, use the search icon to select the configurations relevant to your scope.

Related Information

Create Communication Configurations
View Inbound Configurations
View Outbound Configurations

5 Configuration in SAP S/4HANA

An overview of how to configure integration with S/4HANA system.

Prerequisites

- You are using Convergent Contract Accounting (FI-CAC) or SAP Contract Accounts Receivable and Payable (FI-CAX).
- You have upgraded your system to SAP S/4HANA 2025.
- You have implemented 3764097 to enable all the new features of the SAP Service Cloud Version 2 release.
- In addition, if you are using SAP Contract Accounts Receivable and Payable (FI-CAX), you have implemented 3781845.
- Your administrator has business user access to configure SAP S/4HANA.

5.1 Maintain Trust between SAP Cloud Integration and SAP S/4HANA

Trust relationships enable you to establish secure connections between known servers whose identity can be confirmed by a signed certificate.

Context

You need to download the certificate of SAP Cloud Integration tenant via a web browser and import it to SAP S/4HANA. Google Chrome is used as the default browser.

Procedure

1. Access SAP Cloud Integration using the URL https://%3Ctenant%20management%20node%3E/itspaces
2. In the browser, choose the lock icon and then *Certificate* (Valid) of your secure connection.
3. In the *Certificate* dialog box, go to the *Details* tab.
4. Choose *Copy to File*.

5. In the *Certificate Export Wizard* dialog box, choose *Next*.
6. Choose *Base-64 encoded X.509 (.CER)* and choose *Next*.
   Certificates can be exported in a variety of file formats.
7. Specify the name of the file and save it to the desktop by default. Choose *Next*.
8. Choose *Finish*.

Results

A message tells you that the export was successful.

Upload SAP Cloud Integration Certificate to S/4HANA

Context

Uploading SAP Cloud Integration certificates to SAP S/4HANA is a crucial step in establishing a secure communication setup between the two systems. Certificates are used for authentication, ensuring that the communication between SAP Cloud Integration and SAP S/4HANA is secure and trusted.

Procedure

1. Log on to your SAP S/4HANAsystem as an administrator (**`SAP_BR_ADMINISTRATOR`**).
2. Open *Maintain Certificate Trust List*.
3. Next to the *Certificate Trusts* table, choose + (Add icon).
4. In the *Upload Certificate* dialog box, browse for the SAP Cloud Integration certificate that you've downloaded and choose *Upload*.

Results

A system message informs you that the certificate has been uploaded.

5.2 SAP S/4HANA Web Service Configuration (SOA Management)

This feature is intended for technical system administrators who configure inter-system connections and manage security settings in the system landscape.

To use web services with Web Services Reliable Messaging, you must have the web service runtime configured.

The configuration of the web service runtime is client-specific and must be performed in each productive client.

User Action	Entry Value
Transaction Code	SOAManager

For all the S/4H inbound and outbound web service names, refer the SAP Note

5.2.1 Outbound Web Service Configuration

Procedure

1. Open *Service Administration* tab and browse *Web Service Configuration*. Search for the entity name and complete the steps that follow.

Entity Name	Name
CO_FKK_MDG_CONTRACT_ACCOUNT_BU	ContractAccountBulkReplicateRequest_Out

2. Configure the logical port.

   If the logical port is already configured for a different system then follow the steps in SAP Master Data Integration service Business Partner configuration without using Default Logical Port on SAP S/4HANA and skip below steps.
3. Choose *Create*.
4. Choose *Manual Configuration*.
5. Under *Logical Port Name*, add the logical port name. For example, `LP_<system name to connect to>`.
6. Check the Logical port as default box.
7. Add the port description.
8. Choose *Next*.
9. Under *Consumer Security*, configure authentication and authorization as described for the authorization type in your integration flow:

Integration Flow Authorization Type	
User Role	Choose the User ID/Password option. Add the SCI User ID and password to connect to the SCI tenant.
Client Certificate	Choose **X.509 SSL Client Certificate for Authentication Settings.** Choose an item for your SSL Client PSE of transaction **STRUST**.

10. Choose *Next*.
11. Under *HTTP/ Transport* Settings, in the **URL** field, add the SCI integration flow end point.

    In case you are adding the URL-by-URL components then in the *Protocol* field, add `HTTPS`.
12. Choose *Next*.
13. Under *SOAP Protocol /Messaging*, add the fields as shown:

Field Name	Value
RM Protocol	<SAP RM>
Message ID Protocol	<SAP Message ID>
Data Transfer Scope	<Basic Data Transfer>
Transfer Protocol	<Transfer via SOAP header>

14. Choose *Next*.
15. Choose *Finish*.
16. Repeat the same steps for creating the logical port for the other entity.

> ⓘ Note
>
> The `SAP Note 3253818` provides gives all information about the S/4 web service names, CI integration flow names and service cloud communication configuration details. The attached excel in the note has all the technical information for the integration setup.

5.2.2 Inbound Web Service Configuration

Activate the SAP S/4HANA inbound web service to set up the connection for communication from SAP Cloud Integration to SAP S/4HANA.

Context

You can activate the inbound web services listed below for communication from SAP Cloud Integration to the SAP S/4HANA system by enabling the service definitions in SAP S/4HANA:

- isu_c4c_v2_alerts

- isu_c4c_v2_ami_ping
- isu_c4c_v2_bank_data
- isu_c4c_v2_bank_master
- isu_c4c_v2_bill_amount
- isu_c4c_v2_bill_document
- isu_c4c_v2_bp_data
- isu_c4c_v2_bp_security_dep
- isu_c4c_v2_bpem
- isu_c4c_v2_budget_bill_plan
- isu_c4c_v2_ca_details
- isu_c4c_v2_ca_overview
- isu_c4c_v2_ca_payment_data
- isu_c4c_v2_collections
- isu_c4c_v2_consumption
- isu_c4c_v2_contracts_get
- isu_c4c_v2_corresp_hist_get
- isu_c4c_v2_credit_worthiness
- isu_c4c_v2_device
- isu_c4c_v2_disconn_doc
- isu_c4c_v2_dunning
- isu_c4c_v2_financial_amount
- isu_c4c_v2_financial_doc
- isu_c4c_v2_installment_plan
- isu_c4c_v2_invoices
- isu_c4c_v2_locks
- isu_c4c_v2_manual_posting
- isu_c4c_v2_mdt_ca_crt
- isu_c4c_v2_meter_read
- isu_c4c_v2_move
- isu_c4c_v2_move_in
- isu_c4c_v2_move_in_mdt
- isu_c4c_v2_move_out
- isu_c4c_v2_object_address_get
- isu_c4c_v2_object_search
- isu_c4c_v2_open_item_graph
- isu_c4c_v2_open_items
- isu_c4c_v2_otp_create
- isu_c4c_v2_payment_plan
- isu_c4c_v2_payments
- isu_c4c_v2_premise_bp
- isu_c4c_v2_premise_service
- isu_c4c_v2_promise_to_pay

- isu_c4c_v2_rate_category
- isu_c4c_v2_returns
- isu_c4c_v2_transfer_sec
- isu_c4c_v2_util_mdt_ca_crt
- isu_c4c_v2_write_off
- SALESORDERBULKREQUEST_IN
- CRMS4_IU_CM_PROCESS_SD_MON

Procedure

1. Go to the **SOA Management** (SOAMANAGER) transaction in your SAP S/4HANA system.
2. Choose *Simplified Web Service Configuration*.
3. Enter the web service definition name in the *Search Pattern* field and choose *Go*.
4. In the result list, select the Username/Password (Basic) check box for the web service definition.
5. Choose *Save*.

6 Configuration in SAP Cloud Integration

SAP provides prepackaged, generic integration content called integration flows for the integration between SAP S/4HANA and SAP Service Cloud Version 2 for using SAP Cloud Integration.

> **Note**
>
> You must not make any updates to the standard package delivered by SAP. Doing this will lead to missing of auto-update of the integration package, and you can see an error message stating the failure of uniqueness check. For more details on the error, see [https://me.sap.com/notes/]0003531297.

6.1 Copy Integration Package

SAP Service Cloud V2 / SAP Enterprise Service Management for Financials Customer Care Integration with SAP S/4HANA contains integration flows that need to be copied and configured.

Procedure

1. Log in to the SAP Cloud Integration tenant.
2. From the ☰ (Menu), select *Discover*.
3. Click *Integration*.

   All the SAP delivered integrations packages are listed.
4. Search and select the utilities *SAP Service Cloud V2 Master Data Integration with SAP S/4HANA* package.
5. Click (Copy).
6. Repeat the same steps for the other integration packages.

6.2 Deploy Security Credentials

The communication between SAP Cloud Integration and SAP Service Cloud Version 2 is established via basic authentication over HTTPS. User credential needs to be deployed in the SAP Cloud Integration tenant, which is used for sending messages to SAP Service Cloud Version 2.

Procedure

1. Log in to the SAP Cloud Integration tenant.
2. From the ☰ (Menu), select *Monitor*.
3. Under *Manage Security*, select *Security Material*.
4. From the *Create* dropdown, select *User Credentials*.
5. In *Create User Credentials*, enter the required details.

   Use the credential maintained in *SAP Service Cloud Version 2* > *Settings* > *Integration* > *Communication Systems* > *Inbound*.
6. Click *Deploy* to deploy the user credential artifact.
7. For basic authentication of inbound messages in SAP S/4HANA, repeat the steps for the user authorized to execute messages in the SAP S/4HANA system.

6.3 Configure and Deploy Integration Flows

After copying the integration packages to your workspace, you need to set up and deploy the integration flows.

The integration package is SAP Service Cloud V2/ SAP Enterprise Service Management for Financials Customer Care Integration with SAP S/4HANA.

6.3.1 Configure and Deploy Each Integration Flow Individually

After copying the integration packages to your workspace, you need to set up and deploy the integration flows.

Procedure

1. Log in to the SAP Cloud Integration tenant.
2. Configure the integration flows, from the ☰ (menu), select *Design*.
3. Select the copied integration package *SAP Service Cloud V2 / SAP Enterprise Service Management for Financial Customer Care Integration with SAP S/4HANA*.

All integration flows are displayed on the *Artifacts* tab.

4. Select the required integration flow.
5. From the *Actions* menu, select *Configure*.
6. Configure the sender system (**SAP Service Cloud Version 2**) and the receiver system (**S4H**) details:
   a. In the *Sender* tab, select the *Authorization and User Role* options.
   b. Click the *Receiver* tab and do the following:
      - In *Address*, enter the endpoint address of the receiving SAP S/4HANA system.
        The URL needs to start with a protocol, host, and port name of the S4H system. By default S4 host, port, and client are defined as properties. Replace **{property.S4HostandPort}** and **{property.S4Client}** with your SAP S/4HANA host, port, and client details, for example, http://abc:44300/sap/bc/srt/scs/sap/fkk_esm_intgr_0124?sap-client=001.
        You can locate the inbound endpoint URL for each S4 service in the S4H system by navigating to transaction Tx.SOAMANAGER.

     > ### Note
     > Ensure these properties are retained and deployed when configuring using method 2.

   c. From the *Proxy Type* dropdown, select *On-Premise* (for connectivity via SAP Cloud Connector) or *Internet* (for connectivity via reverse proxy based infrastructure).
      Delete the property `${property.LocationID}` and update your Location ID if you are using one; otherwise, leave this parameter blank.

     > ### Note
     > You can connect multiple Cloud Connectors to a sub account, provided each has a unique location ID. The location ID identifies which Cloud Connector is used to establish the connection. By default, the location ID is an empty string, indicating the Cloud Connector connected without a specific location ID.

   d. From the *Authentication* dropdown, select *Basic*.
   e. In the *Credential Name* field, enter the name of the credential deployed in security artifacts.
   f. Repeat the same procedure for all the integration flows that have the same sender as SAP Service Cloud Version 2 and the receiver as **S4H** systems.
7. Repeat the steps from 3 to 6 for the integration package *SAP Service Cloud V2 for Utilities Premise Integration with SAP S/4HANA*.

Related Information

SAP S/4HANA Web Service Configuration (SOA Management) [page 11]
This feature is intended for technical system administrators who configure inter-system connections and manage security settings in the system landscape.

6.3.1.1 Configure Getting Utilities Details for Service Order Creation

Retrieve utilities-related data from SAP S/4HANA.

Prerequisite: The integration flow Replicate Service Order from SAP S/4HANA to SAP Service Cloud Version 2 is configured and deployed.

The integration flow Get Utilities Details for Service Order Creation in the integration package SAP Service Cloud V2 for Utilities Premise Integration with SAP S/4HANA is a part of the main integration flow, and acts as an extension to the integration flow `Replicate Service Order from SAP S/4HANA to SAP Service Cloud Version 2`.

It is triggered when the configurable parameter Is Application Extension Enabled is set to true.

The purpose of this integration flow is to run the SAP S/4HANA `isu_c4c_v2_object_search` service to retrieve utilities-related data, for example, premise ID. The retrieved information is then passed back to the main integration flow Replicate Service Order from SAP S/4HANA to SAP Service Cloud Version 2, in the integration package SAP Service Cloud Version 2 Integration with SAP S/4HANA. This handles the replication of service orders from SAP S/4HANA to SAP Service Cloud Version 2.

Related Information

Integrate SAP Service Cloud Version 2 with SAP Field Service and Asset Management
Configurations in Cloud Integration

6.3.2 Configure and Deploy all Integration Flows Concurrently

This approach can significantly reduce implementation time.

Before deploying all integration flows, deploy the *Automated Integration Flow Deployment Configurator* first by providing the configurable parameters. This integration flow saves the parameters as variables, which are then used by other integration flows at runtime.

> **Note**
>
> - The variable expires after a retention period of 400 days.
> - The retention period begins once the integration flow containing the variable is successfully processed for the first time.
> - The retention period for the variable is extended with each successful processing of the integration flow.

1. Log in to the SAP Cloud Integration tenant.
2. To configure the integration flows, from the ☰ (menu), select *Design*.

3. Select the copied integration package SAP Service Cloud Version 2 for Utilities Contract Account Integration with SAP S/4HANA.
   All integration flows are displayed on the *Artifacts* tab.
4. Select the *Automated Integration Flow Deployment Configurator* artifact.
5. From the *Actions* menu, select *Configure*.
6. In the *Timer* tab, make the selections as follows:
   - *Frequency*: Select *Repeat Yearly*.
   - *Start Date and Time*: Select a time shortly after the planned deployment, for example, 2 minutes after deployment.
   - *End Date and Time*: Select *2050* (marked by default, you can modify it if required).
   - *Time Zone*: Select as per your time zone.

   > ⓘ Note
   >
   > The *Timer* tab is available only in versions 1.0.1 and above. In the older versions, a timer is set to deploy the integration flow annually by default, but you can also schedule it according to your convenience.
7. In the *More* tab, enter the fields accordingly:
   - CloudConnectorLocationID
   - S4Client
   - S4Host and Port
8. Select **Deploy**.
   This saves the configurable parameters as variables, which will be utilized at runtime once all the integration flows are deployed.
9. Now select rest all other integration flows expect *Automated Integration Flow Deployment Configurator*.
10. From the *Actions* menu, select *Configure*.
11. Click **Deploy All**.
    All integration flows will be deployed simultaneously.

6.4 Configure Master Data Replication

Procedure

1. Select the integration flow *Replicate Contract Account from S/4HANA* that you want to configure.
2. Select the details for the sender system (S4H), and the receiver systems (SAP Sales Cloud and SAP Service Cloud Version 2 and S4H).

   In the *Sender* tab, select the *Authorization* and the *User Role* options.
3. Choose the *Receiver* tab and create the following entries:
   a. Host: SCV2 tenant's URL (without https://)
   b. Port: 443

4. Choose proxy type as *Internet*.
5. Choose *Allow Chunking*.
6. Select *Basic Authentication* from dropdown.
7. Provide the credential name which is deployed under security artifacts.
8. Choose *Save*.
9. Choose *Deploy*.

7 Configuration in SAP Cloud Connector

SAP Cloud connector allows SAP BTP applications to securely access remote services that run on the Internet or SAP S/4HANA.

Context

If you already have a reverse proxy setup, you can reuse it. Refer to Outbound/On-Premise: Reverse Proxy or SAP Cloud Connector to determine the best option for your specific situation.

This section describes the configuration in SAP Cloud Connector.

Procedure

1. Log on to the SCC administration UI.
2. Select the relevant sub account.
3. Choose *Cloud to SAP S/4HANA* under the sub account menu.
4. Under *Access Control* tab select + (Add) button in *Mapping Virtual To Internet System* section.

Field Name	Entry Value
Back-end Type	ABAP System
Protocol	HTTPS
Internal Host	ldciqvp.devsys.net.sap (example)
Internal Port	44300
Virtual Host	qvp (example)
Virtual Port	443
Principal Type	None
Host in Request Header	Use Virtual Host
Description	(Not mandatory)
Check Internal Host	True

5. After each step select the *Next* button and then *Finish*.
6. Select the newly created entry.
7. On the Resources of qvp:443 (example) section select + (Add) button.

Field Name	Entry Value
URL Path	/sap/bc/srt/
Activate	True

8. Save the entry.
9. To check if the configuration was done correctly follow these steps:
    a. Open SAP BTP Cockpit.
    b. Navigate to the *Connectivity* tab.
    c. Select *Cloud Connectors*.
    d. Check if the new entry is available under *Exposed Back-end System* section.

Related Information

https://help.sap.com/docs/cloud-integration/sap-cloud-integration/connecting-customer-system-to-cloud-integration?version=Cloud

8 Replication of Master Data Initial Load

In the following sections, we describe the replication of contract account, utilities technical master data (Connection object and premise), and business partner.

Procedure

1. Load contract account replication
2. Load technical object replication.
3. Load business partner and business partner replication.

> Note
>
> - Start loading data in small chunks, check if they are loaded successfully, and then later increase the size of the data.
> - After loading the data, go to *SAP Service Cloud Version 2* > *Integration* > *Message Monitoring* and check the status of the operation.

8.1 Setting Up SAP S/4HANA for Contract Account Replication

Set up SAP S/4HANA to replicate contract accounts to SAP Service Cloud Version 2.

8.1.1 Creating the Outbound Service for Contract Accounts in the SOA Manager Tool

You create the contract account outbound service in the SOA Manager tool for the data exchange from SAP S/4HANA to SAP Service Cloud Version 2.

Procedure

1. Use transaction **SOAMANAGER** from the S/4HANA system.
2. Under the *Service Administration* tab, choose *Web Service Configuration*.

3. Search for the object name **`CO_FKK_MDG_CONTRACT_ACCOUNT_BU`**.
4. Choose the internal name of the object.
5. Choose *Create* and then select *Manual Configuration*.
6. Configure the logical port name as follows and then choose *Next*.
   a. In the *Logical Port Name* field, enter the logical port name of the system to which you're connecting.
   b. Select the *Logical Port is Default* checkbox.
   c. Enter a description such as **`Contract Account Replication`**.
7. Configure the consumer security with one of the following forms of authentication and then choose *Next*:

   - **Basic Authentication**
     Enter the user ID and password that you use to connect to your SAP Cloud Integration tenant.

     > ⓘ Note
     >
     > By default, the *User ID / Password* radio button is selected.

   - **Certificate-Based Authentication**
     1. Select the *X.509 SSL Client Certificate* radio button.
     2. Choose the value help of the *SSL Client PSE of transaction STRUST* field.
     3. Choose the PSE in which the client certificate that was issued by Certification Authority (CA) and supported by SAP Cloud Integration is stored.
8. Configure the HTTP settings as follows and then choose *Next*.
   a. In the *URL* field, enter the runtime URL from SAP Cloud Integration and append the following string to it: **`/cxf/Utilities/contractAccountReplication`**.
   b. Enter a name such as **`Proxy`** for the proxy host.
   c. Enter a port number such as **`8080`** for the proxy host.
9. Configure the SOAP protocol as follows and then choose *Finish*.
   a. In the *RM Protocol* dropdown menu, select *SAP RM*.
   b. In the *Message ID Protocol*, select *SAP Message ID*.
   c. In the *Data transfer scope* dropdown menu, select *Basic Data Transfer*.

8.1.2 Configuring the Data Replication Framework for Contract Account Replication

You configure the data replication framework for contract account replication.

Procedure

1. Use transaction **`DRFIMG`** in the S/4HANA system.
2. Navigate to Customizing for *Define Technical Settings for Business Systems* under *Data Replication* > *Define Custom Settings for Data Replication* > *Define Technical Settings*.
3. Add a new entry for the business system of SAP Marketing Cloud and then save your changes.

4. Select the business system and then double-click the *Define Bus. Systems, BOs* node in the dialog structure.
5. Configure a new entry as follows and then save your changes.
    a. In the *BO Type* field, enter **DRF_0036**.
    b. Select the *System Filter* checkbox.
6. Select the business object type then double-click the *Define Bus. Systems, BOs, Communication Channel* node in the dialog structure.
7. In the *Communication Channel* dropdown menu, select *Replication via Services* and then save your changes.
8. Navigate to Customizing for *Define Replication Models* under *Data Replication* > *Define Custom Settings for Data Replication*.
9. Configure a new entry as follows and then save your changes.
    a. In the *Replication Model* field, enter a name.
    b. Enter a description.
    c. In the *Log Days* field, enter a value such as **50**.
10. Select the replication model and then double-click the *Assign Outbound Implementation* node in the dialog structure.
11. Configure a new entry as follows and then save your changes.
    a. In the *Outbound Implementation* field, enter **DRF_0036_3**.
    b. In the *Sequence* field, enter **1**.
12. Select the outbound implementation and then double-click the *Assign Target Systems for Repl. Model / Outb.Impl* node in the dialog structure.
13. Add a new entry using the business system that you created in step 3 and then save your changes.
14. Select the business system and then double-click the *Assign Outbound Parameter* node in the dialog structure.
15. Configure a new entry as follows and then save your changes.
    a. In the *Outbound Parameter* field, enter **PACK_SIZE_BULK**.
    b. In the *Outbound Parameter Value* field, enter a value such as **20**.
16. Double-click the *Define Replication Model* node in the dialog structure.
17. Select the replication model and then choose *Activate*.

8.1.3 Configuring the Delta Replication of Contract Accounts

The delta replication of contract accounts uses a function module to monitor for events that are triggered by the creation of or an update to a contract account and then sends requests for the data to be replicated.

Procedure

1. Use transaction **FQEVENTS** in the S/4HANA system.
2. In the *Event* column, double-click *1031*.

3. Choose the *Function Modules* tab.
4. In the *Installation-Specific Function Modules or Cloud BAdI* section, choose *Change Customizing*.
5. In the *Active Module* field, add a new entry for the **FKK_MDG_OUTBOUND** function module and then save your changes.

8.2 (Deprecated) Technical Master Data Replication

Use the Technical Objects Replication Report: `ISU_C4C_TECH_OBJECTS_REPL` to replicate the technical object to the cloud solution.

Configuration in S/4HANA System

To run the `ISU_C4C_TECH_OBJECTS_REPL` report in the S/4HANA system, you need to create logical ports with transaction code `SOAMANAGER` in order to send web service calls to the PI or other middleware system. Follow these configuration steps:

1. Go to transaction code `SOAMANAGER` in the S/4HANA system.
2. Choose *Service Administration* as *Web Service Configuration*.
3. Search for consumer proxy `CO_UTILITIES_TECHNICAL_OBJEST`.
4. Choose *Create a Manual* configuration.
5. Enter information based on your middleware configuration. Here is an example value that needs to be adjusted according to your configuration. **Example URL Path**: https://{{cloudintegrationhost}}/cxf/Utilities/PremiseReplication.
6. To confirm that the logical port was created and configured correctly, ping the Web service. If the ping is successful, a confirmation message appears.

Report for Initial Load - ISU_C4C_TECH_OBJECTS_REPL

Schedule periodic runs for the initial load report S/4HANA system to transfer the utilities technical master data to the cloud solution. This report offers a single interface to replicate the following objects:

- Premise
- Connection Object

Follow these steps to run the report:

1. In the dialog screen for *Utilities Technical Objects Replication*, select and specify value in one of the fields- *Premise ID* or *Connection Object ID*.
2. Specify the *Logical Port ID*.

> **Note**
>
> Language key is set to *sy-langu* if empty. Also the *From-Date* is set to today's date if empty. And the *To-Date* will be set to 31.12.9999 if empty.

3. Specify a number for the field *Maximum Size*. With the field “MaximumSize", you can control how many technical master data is sent in a single message. The web service can upload several Connection Object IDs, Premises, and PoDs at once. Technical Master Data (Connection Object, Premise, and Point of Delivery) will be distributed accordingly.

Delta Replication- Connection Object

Create or change of a connection object triggers delta replication. Follow these steps to trigger a delta replication for connection object data:

1. In the transaction code `SM30`, enter the table / view `SWFDVEVTY2`.
2. Add the entries for the events *CHANGED* and *CREATED* for the object `CONNOBJ`.

> ⓘ Note
>
> Creating or changing a connection object triggers the delta replication for connection objects.

Delta Replication- Premise

Follow these steps to trigger a delta replication for premise data:

1. In the transaction code `SM30`, enter the table / view `SWFDVEVTY2`.
2. Add the entries for the events *CHANGED* and *CREATED* for the object `PREMISES`.

> ⓘ Note
>
> Creating or changing a connection object triggers the delta replication for premises.

> ⓘ Note
>
> Along with the existing fields the *District* field is enabled for replication under the Premise.

Extensibility for Initial Data Replication Report and for Delta Replication

A Business Add-In (BAdI) to modify the standard implementation for the initial data replication report and for the delta replication runs.

- **Enhancement Spot**: `COD_UTIL_TECH_OBJS_REPLICATION`
- **BAdI**: `COD_UTIL_TECH_OBJS_REPLICATION`
- **Interface**: `IF_COD_UTIL_TECH_OBJS_REPL`
- **Methods**: `CONN_OBJ_ADJUST_DATA`, `PREMISE_ADJUST_DATA`,`TMD_ADJUST_DATA`, `SENDER_ADJUST_DATA`, `RECEIVER_ADJUST_DATA`

8.3 Technical Master Data Replication

This data includes connection objects and premises replicated to the SAP Service Cloud Version 2.

The old procedure has been deprecated, so please follow the latest setup instructions for configuration.

8.3.1 Integration in SAP Service Cloud Version 2 for Utilities

Integrate the SAP Service Cloud Version 2 for utilities with S/4HANA to replicate the premise master data.

Procedure

1. Go to *User Menu* > *Settings* > *All Settings* > *Integration* > *Communication Configuration*.
2. Select *Integrate Service Cloud Utilities Master Data with SAP S/4HANA* module.
3. Delete the *Replicate Utilities Technical Master Data from SAP S/4HANA* configuration (if existing) in the Inbound Configuration section.
4. Search and add *Replicate Premise from SAP S/4HANA* configuration.

8.3.2 Integration in SAP S/4HANA

Integrate the SAP S/4HANA with SAP Service Cloud Version 2 for utilities to replicate the premise master data.

Procedure

1. Maintain the system level to the minimum required `IS-UT 808 SP0` package and implement below OSS note **3484932**.
2. Set up SOA Manager configuration for the service **CO_ISU_C4C_V2_PREM_CONNOBJ_REP**.
3. Implement the BAdI *ISU_C4C_V2_TECH_OBJS_REPL* and pass the receiver system data using *RECEIVER_ADJUST_DATA*.

   Optional: The customer can switch to the sender system by using the SENDER_ADJUST_DATA BAdI.
4. The initial load report *ISU_C4C_V2_TECH_OBJECTS_REPL* is created in the backend to replicate the premise data.
5. Maintain a workflow configuration for delta replication using event class *CL_ISU_C4C_V2_TECH_OBJ_REP*.
   a. Go to SWE2 T.Code
   b. Maintain the configuration for *CONNOBJ* and *PREMISE* business objects.
   c. For Connection Object maintain the change event as:

- Enter the receiver type as **ISU_C4C_V2_TECHOBJ_REP**.
- Maintain class name as **CL_ISU_C4C_V2_TECH_OBJ_REP**.
- Enable the **Linkage Activated** check box.
- Save the configuration.

> ⓘ Note
>
> For connection object create event is not required.

d. For Premise business object maintain create and change events as:

- Enter the receiver type **ISU_C4C_V2_TECHOBJ_REP**.
- Maintain class name as **CL_ISU_C4C_V2_TECH_OBJ_REP**.
- Enable the **Linkage Activated** check box.
- Save the configuration.

6. To extend the replication, implement the BAdI's for extensibility **isu_c4c_v2_tech_objs_repl**.

> ⓘ Note
>
> The BAdIs implemented in V1 cannot be used for the V2 new services. Therefore, a new BAdI must be implemented for V2.

7. In new service, customer can extend the proxy structures **Premise_ConnOBJ** in S/4HANA and re-generate the service using SProxy(T.Code) - **UtilitiesPremiseConnectionObjectReplicateRequest_Out** under the namespace http://sap.com/xi/C4CS4/IS-UT.
8. Implement appropriate BAdI **isu_c4c_v2_tech_objs_repl** method to map the custom fields.

Related Information

SAP S/4HANA Web Service Configuration (SOA Management) [page 11]
This feature is intended for technical system administrators who configure inter-system connections and manage security settings in the system landscape.

8.3.3 Integration in CPI

Integrate the CPI with SAP Service Cloud Version 2 for utilities to replicate the premise data.

Procedure

1. Update the CPI package *SAP Service Cloud V2 for Utilities Master Data Integration with SAP S/4HANA* to latest version.
2. Undeploy the existing **Replicate Utilities Technical Master Data from SAP S4HANA** and **Replicate Utilities Technical Master Data from S4HANA Extensibility** integration flows configured in your system.

3. Configure and deploy the latest integration flow **Replicate Utilities Premise Connection Object from S4HANA**.

   For more information about configure and deploy integration flows refer the related link section.

Related Information

Configure and Deploy Integration Flows [page 16]
After copying the integration packages to your workspace, you need to set up and deploy the integration flows.

8.4 Data Replication for Point of Delivery

This data includes connection objects and premises replicated to the SAP Service Cloud Version 2.

8.4.1 Setting Up SAP S/4HANA Utilities for POD Replication

Set up SAP S/4HANA Utilities to replicate Points of Delivery (POD) to SAP Service Cloud Version 2.

Context

Create the POD outbound service in the SOA Manager tool for the data exchange from SAP S/4HANA Utilities to SAP Service Cloud Version 2.

Procedure

1. Use transaction **`SOAMANAGER`** from the SAP S/4HANA system.
2. Under the *Service Administration* tab, choose *Web Service Configuration*.
3. Search for the object name **`CO_ISU_C4C_V2_POD_REPL`** or **`UtilitiesPointOfDeliveryReplicateRequest_Out`**.
4. Choose the internal name of the object.
5. Choose *Create* and then select *Manual Configuration*.
6. Configure the logical port name as follows and then choose *Next*.
   a. In the *Logical Port Name* field, enter the logical port name of the system to which you're connecting.
   b. Select the *Logical Port as Default* checkbox.

c. Enter a description such as **`Utilities POD Replication`**.

7. Configure the consumer security with one of the following forms of authentication and then choose *Next*.
   - **Basic Authentication**
     Enter the user ID and password that you use to connect to your Cloud Integration tenant.

     > **Note**
     >
     > By default, the *User ID / Password* radio button is selected.

   - **Certificate-Based Authentication**
     1. Select the *X.509 SSL Client Certificate* radio button.
     2. Choose the value help of the *SSL Client PSE of transaction STRUST* field.
     3. Choose the PSE in which the client certificate that was issued by Certification Authority (CA) and supported by Cloud Integration is stored.
8. Configure the HTTP settings as follows and then choose *Next*.
   a. In the *URL* field, enter the runtime URL from Cloud Integration and append the following string to it: /cxf/Utilities/PremiseAndConnectionObjectReplication.
   b. Enter a name such as Proxy for the proxy host.
   c. Enter a port number such as 8080 for the proxy host.
9. Configure the SOA protocol as follows and then choose *Finish*.
   a. In the *RM Protocol* dropdown menu, select *SAP RM*.
   b. In the *Message ID Protocol*, select *SAP Message ID*.
   c. In the *Data transfer scope* dropdown menu, select *Basic Data Transfer*.

8.4.2 Integration in SAP S/4HANA

Understand the steps required to configure the Data Replication Framework (DRF) for Point of Delivery (POD) replication.

Procedure

1. Access DRF customizing using the transaction `DRFIMG`.
2. Navigate to *Data Replication* > *Define Custom Settings for Data Replication* > *Define Technical Settings* > *Define Technical Settings for Business Systems*.
3. Define the business system.
   a. Add a new entry for the business system **SAP Marketing Cloud** and save your changes.
   b. Select the business system and double-click *Define Bus. Systems, BOs* in the dialog structure.
4. Define the business object.
   a. Create a new entry with the *BO Type* as **`ISU_POD`** and *System Filter* as **`Selected`**, and save it.
   b. Select the business object type and double-click Define Bus. *Systems, BOs, Communication Channel*.
   c. From the *Communication Channel* dropdown, select *Replication via Services* and save.

5. Define the replication model.
    a. Navigate to *Data Replication* > *Define Custom Settings for Data Replication* > *Define Replication Models*.
    b. Create a new replication model with the following values:

Field	Value
Replication Model	Enter a suitable name.
Description	Enter a meaningful description
Log Days	50 (modify as required)

6. Assign outbound implementation.
    a. Select the replication model and double-click *Assign Outbound Implementation*.
    b. Create an entry for the *Outbound Implementation* as ISU_POD, and set the *Sequence* as 1.
7. Assign the target system.
    a. Select the outbound implementation and double-click *Assign Target Systems for Repl. Model / Outb. Impl.*
    b. Add a new entry using the business system created in step 3 and save.
8. Maintain outbound parameters.
    a. Select the business system and double-click *Assign Outbound Parameter*.
    b. Create a new entry with the *Output Parameter* as PACK_SIZE_BULK and the value as 20, or as required.
9. To activate the replication model, double-click *Define Replication Model*, select the replication model and choose *Activate*.
10. To enable delta replication, you must maintain the required workflow configuration.
    a. Go to transaction `SWE2`.
    b. Maintain the workflow event linkage for the `ISUPOD` business object.
    c. Ensure the relevant workflow for the event class CL_ISU_C4C_V2_POD_DRF_OUT is correctly configured and active for delta replication processing.

8.4.3 Maintain IMG Documentation

Procedure

1. Enhance the IMG structure.
    a. Go to transaction `S_IMG_EXTENSION` (Edit IMG Structure).
    b. Enter `SIMG_ISU` as the IMG Structure.
    c. Enter `ISU_C4C_S4H_UTI` as the Enhancement ID.
    d. Choose *Enhance Structure*.
2. Create the IMG node.

a. Insert a subnode under *Cloud for Utilities*.
b. Enter the document name and attach the document in the *Document* tab.

3. Fill the required details in the *Attributes* and *Customizing Activity* tabs.
4. In the *Document* tab, enter the following information: Using this configuration step, you can activate Point of Delivery replication across all services for the Utilities solution in SAP Service Cloud Version 2. This configuration enables the required web services to replicate Point of Delivery data from SAP S/4HANA Utilities to the solution
5. Save the main IMG node.

   You can use the following BAdI enhacements:

   - `MDG_ID_MAPPING_API` to modify the local system determination during Data Replication Framework processing.
   - `ISU_C4C_V2_TECH_OBJS_REPL` to modify sender and receiver information during Point of Delivery replication.
   - `ISU_C4C_V2_REPLICATION_CONF` to dynamically change the logical port used for outbound replication.

8.5 Load Business Partner and Business Partner Relationships

For business partners you must use data replication framework.

Procedure

1. Run the *DRFOUT* transaction code.
2. In *Replication Model*, select your replication model.
3. In *Outbound Implementation*, select *986_3*.
4. In *Replication Mode*, select *Manual*.
5. Click *Manual replication Filter Criteria*.
6. Select the required business partners and relations.

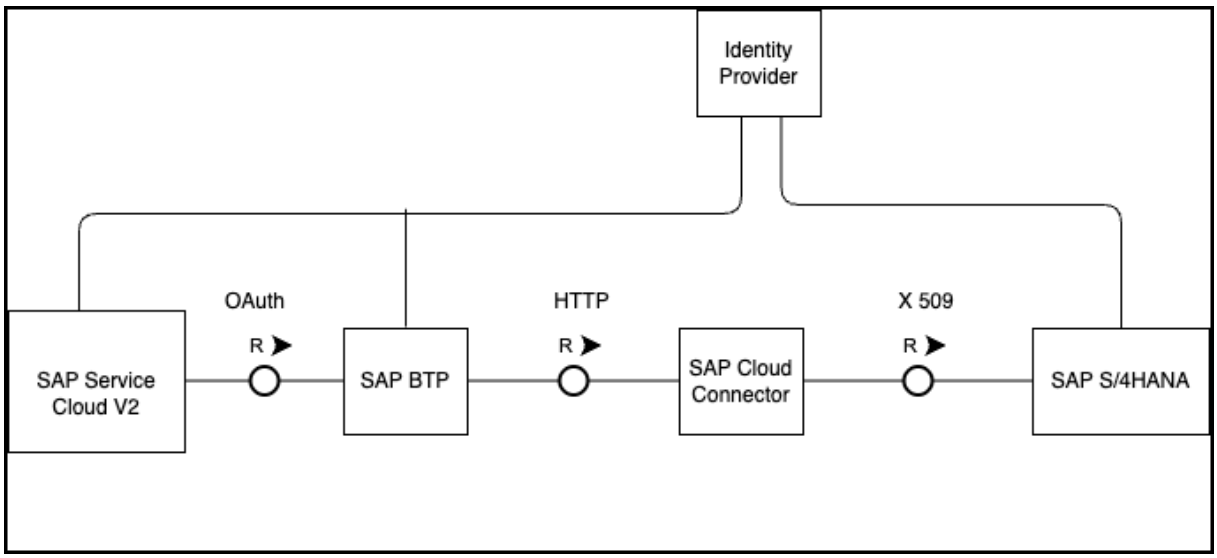
9 Principal Propagation

Principal propagation is a secure, Single Sign-On mechanism that forwards an end-user's identity (principal) from a cloud application to an on-premise back-end system without changing it.

In this section, you can understand how to use principal propagation in SAP Service Cloud V2.

Principal propagation is supported only for synchronous transactions (SAP Service Cloud V2 to SAP S/4HANA) and is not supported for asynchronous integrations such as contract account and premise replication.

System Landscape



- Sender: SAP Service Cloud V2
- Middleware: SAP BTP Cloud Integration (CPI)
- Connector: SAP Cloud Connector (SCC)
- Receiver: SAP S/4HANA (On-Premise)

Authentication Mechanism

- OAuth 2.0 SAML Bearer Assertion from SAP Service Cloud V2 to SAP BTP
- X.509 Certificate Propagation from SAP BTP to SAP S/4HANA via SAP Cloud Connector

The scenario follows an Identity Provider (IdP) initiated SAML SSO pattern, where SAP Service Cloud V2 acts as a Trusted SAML Issuer and performs a SAML to XSUAA token exchange based on RFC 7522 (SAML 2.0 Bearer Assertion).

Related Information

OAuth 2.0 SAML Bearer Assertion Authentication
Inbound Web Service Configuration
Setting Up Principal Propagation (Example Scenario)

9.1 Configure Trust

To set up principal propagation, trust configuration must be established between SAP Service Cloud V2 and the SAP BTP subaccount where the SAP CPI tenant resides.

Context

Based on internal testing, this method has been verified to work as expected. You may choose alternative methods depending on your project requirements.

Procedure

1. Create a key-pair using OpenSSL (self-signed certificate is also accepted).
2. Prepare an `IdPSSODescriptor` metadata XML file.
3. Upload the metadata XML as a Trusted Identity Provider in the SAP BTP subaccount.

Related Information

Set Up Trust

9.2 Configure Principal Propagation

Principal Propagation allows the identity of a logged-in user to be securely forwarded through multiple systems without using technical users. The objective of this scenario is to propagate the logged-in SAP Service Cloud V2 user identity to SAP S/4HANA without using a technical user for business authorization.

Context

In the following example setup, the principal of the outbound user is forwarded to SAP Cloud Connector, and from there to the back-end receiver system. We assume the following:

- In this example, the authentication option OAuth is used (using OAuth SAML 2.0 Bearer Assertion) for outbound communication from the sender to SAP BTP.
- SAP S/4HANA is the receiver system.
  The receiver system is connected to SAP BTP through SAP Cloud Connector.

  > ⚠ Caution
  >
  > Using SAP Cloud Connector is mandatory when configuring principal propagation.

- To establish an outbound connection (SAP BTP to SAP Cloud Connector), an adapter that supports principal propagation is used (for example, the SOAP receiver adapter).

Procedure

1. Create a communication system in SAP Sales and Service Cloud V2, and use OAuth SAML 2.0 bearer assertion as the authentication method.
2. In the receiver channel of the integration flow, enable *Principal Propagation* as the *Authorization* method.
3. Prepare SAP Cloud Connector to support principal propagation with X.509 certificates (for the communication with the receiver system).

   You need a certificate chain with at least one intermediate certification authority. The intermediate certification authority signs a short-lived certificate, which is used for principal propagation. Use the user name (associated with the identity to be propagated) as the subject common name (subject CN) of this certificate.

   SAP Cloud Connector forwards the identity (to be propagated) in a short-living X.509 certificate in HTTP header SSL_CLIENT_CERT.
4. In SAP Cloud Connector, configure the trust relationship with the SAP BTP application.
5. Configure the receiver system. You need to do the following:
   a. Configure the receiver system to trust the certificate of the SAP Cloud Connector.
   b. Configure the Internet Communication Manager (ICM) to trust the system certificate for principal propagation.
   c. Map the short-living certificate (from SAP Cloud Connector) to the user (whose identity is being propagated).

Related Information

Setting Up Principal Propagation (Example Scenario)
OAuth 2.0 SAML Bearer Assertion Authentication

10 Extensibility

Extensibility allows you and your partners to extend the software platform without having to modify the original codebase. So, you can add to the base functionality, thereby offering new capabilities and outputs.

1. Create the custom field for the entity in SAP Service Cloud V2 and generate OpenAPI schema.
2. Create the custom field in SAP S/4HANA and generate the WSDL.
3. Integrate the OpenAPI schema generated from SAP Service Cloud V2 and the WSDL file from SAP S/4HANA.

10.1 Create Extension Fields

Extension fields are additional fields that you, as an administrator, can add to a cloud solution from SAP.

Context

Extension fields provide the means to store additional data for specific use cases beyond what is natively present in the solution. You can create these fields for a screen that has been enabled for extension fields.

Procedure

1. Navigate to *User Menu* > *System Settings* > *All Settings* > *Extensibility* > *Extensibility Administration*.

   The system opens a new screen and displays a list of all business entities in the left panel. These entities are grouped by services.
2. Select the *Extendable* checkbox to view the entities that can be extended.
3. Drill down to the entity that you want to add an extension field to, and select it. The system opens the corresponding worklist.
4. In the *Custom Fields* tab, select + *(Add)* to open the *Create Fields* quick create screen.
5. Enter the *Label*. The system automatically generates the *Technical Name*. You can edit the technical name if the system displays a duplication error. Note that technical names are unique and not case-sensitive.
6. Select the *Data Type* and the corresponding *Data Format*.
7. (Optional) If the *Data Type* is *String* and the *Data Format* is *UUID*, you can make this field a standard value help option. To do so, select the *Standard Value Help* checkbox, and choose the desired value help from the dropdown list to associate it with the extension field.

> **Note**
>
> Standard Value Help has restricted availability and is available only for Sales Quotes and Sales Orders. To request access to these features, you must create a case with the component CEC-CRM-CZM-EXT.

8. (Optional) If the *Data Type* is **String** and the *Data Format* is **Code**, you can import and export static code list.

> **Note**
>
> Before you export or import, add at least one code list value and save the extension field.

- To export, choose *(Export)*. In the *Export* screen, select the file type as either *Excel* or *CSV* and choose *Export*.
- To import, choose *(Import)*.
  1. In the *Import* screen, you can select *Create* for creating new values or *Update* for updating the existing values. When you are updating the existing file, the system replaces the entire list of values. Ensure that you export the existing values, make the required updates in the file, and then import the file back.
  2. Select the file type as either *Excel* or *CSV*.
  3. Browse for the prepared data file you want to upload.
  4. Choose *Import* and save your changes.

9. (Optional) If you want to map this field with an existing field in a different entity, turn on the *Default Mapping* switch and select *(Add from List)*.

   Default mapping lets you copy a field value from a related business entity into this custom field. When you use default mapping, the value of the selected source field is copied into this custom field for the first time when the record is created or when the entity association is set.

   The available source fields in the list are filtered based on the Data Type and Data Format you selected. From the list, select the source field from either the Direct References or Indirect References section. The behavior of the custom field after the initial copy depends on the reference type you choose.

   - *Direct References*: Direct references are fields from an entity directly associated with the current entity, such as the Account associated with a Lead. When you select a field from Direct References, its value is copied to the custom field when you first set the association. The value updates only when the associated entity itself is changed on the record. You can update the value manually in this custom field. If you manually update the value of this custom field, it does not affect the source entity's field value.

     > **Example**
     >
     > A Lead is associated with Account A, which has a field Region with value **Europe**. A custom field on the Lead is mapped to the Region field via direct reference and shows **Europe**. If you update the Lead to reference Account B, which has Region **Asia**, the custom field automatically updates to **Asia**.

   - *Indirect References*: Indirect references are fields from entities that have an indirect relationship with this entity, typically used in follow-up or create-with-reference scenarios, such as a Sales Order created with reference to an Opportunity. When you select a field from Indirect References, the value is copied from the source field into this custom field once, at the time the record is created. After the record is created, the value remains unchanged regardless of any subsequent changes to the source entity.

There is no automatic update trigger. You can manually update the value in this custom field without affecting the corresponding field value in the source entity.

> **Example**
>
> A Sales Order is created from an Opportunity. A custom field on the Sales Order is mapped to a field on the Opportunity via indirect reference. The Opportunity's field value is copied into the Sales Order custom field when the Sales Order is created. If you later update the Opportunity, the Sales Order custom field value is not affected.

10. (Optional) If you want to mark the custom field that you are creating as a custom reference, enable the *Custom Reference Attribute* switch. Based on the data type and data format you enter, you must select the attributes. Any changes made to this standard field is automatically updated in the reference attributes across the referenced entities. You can select both standard fields and extension fields.

> **Note**
>
> The following field types are not supported:
>
> - Country/Region
> - Amount
> - Quantity
> - Extension fields of standard value help
> - Custom code lists

11. Select *Apply* and save your changes.

    The extension field is now created and is listed under the *Custom Fields* tab.

> **Note**
>
> You can create up to 150 extension fields for root or main entity and up to 50 extensions fields for a child entity.

Related Information

Extension Field Data Types [page 41]

Extension Field Data Types refer to the various formats or structures of data that can be used when creating custom fields (extension fields) in an application. These data types define the kind of data the field can store, such as text, numbers, dates, boolean values, amounts , ensuring that the data entered aligns with the intended use and system requirements.

Field Attributes [page 43]

Field attributes, also known as column attributes or field properties, are characteristics or settings associated with a database field (column) that define various aspects of how data is stored, displayed, and processed within a database. These attributes help ensure data integrity, enforce constraints, and control how the data is used.

Delete Extension Fields

10.1.1 Extension Field Data Types

Extension Field Data Types refer to the various formats or structures of data that can be used when creating custom fields (extension fields) in an application. These data types define the kind of data the field can store, such as text, numbers, dates, boolean values, amounts , ensuring that the data entered aligns with the intended use and system requirements.

Extension Field Data Types

Data Type	Data Format	Description
`STRING`	`STRING` (Default)	This data type is used for storing alpha-numeric characters. It can be a single line or a multiline text field.
	`UUID`	A UUID (Universally Unique Identifier) is a 128-bit identifier used for unique resource identification, typically represented as a 36-character string.
	`EMAIL`	Email data types are designed for storing email addresses and often include validation checks.
	`CODE`	`CODE` data type is used for code list. Fixed code lists are defined via Enum options. Dynamic code lists are defined via reference to the corresponding entity.
	`TOKEN`	A `TOKEN` represents a small piece of data, often a single word, number, or symbol used as a fundamental unit in programming or language processing.
	`URI`	A URI (Uniform Resource Identifier) is a text string used to identify a resource, typically consisting of a scheme, authority, path, query, and fragment components. Example: `https://www.example.com/resource?param=value#section`.
`NUMBER` **Note** Do not store ID fields as the number data type. Use the string data type for ID values. Even if the IDs consist only of numbers, we recommend that you store them as string type, especially when number formatting is not required.	`FLOAT` (Default)	Number data types can include integers, decimals, or floats. They are used for storing numeric values. Use `FLOAT` when memory conservation is critical, and you can tolerate lower precision
	`DOUBLE`	The `DOUBLE` data type is used to represent real numbers with a high level of precision. Double provides a higher level of precision compared to the `FLOAT` data type. It can represent approximately 15-16 decimal digits accurately.

Data Type	Data Format	Description
BOOLEAN		Boolean fields can have only two values - true or false. They are used for binary data or simple yes/no choices.
OBJECT		OBJECT is a versatile and often generic data type that can represent a wide range of values, data structures, or entities.
	AMOUNT	AMOUNT stores numerical values that represent quantities, prices, costs, or any other kind of monetary or non-monetary amounts.
	QUANTITY	QUANTITY typically refers to a numerical value that represents a count, measurement, or amount. Such values are generally represented using standard numeric data types, which can be either integer or floating-point types, depending on the nature of the quantities being measured.
INTEGER	INT32 (Default)	The INT32 data type can store a range of integer values, both positive and negative, within the constraints of a 32-bit binary representation. Int-32 is more memory-efficient than larger integer types like int-64, making it a good choice for applications where memory optimization is a concern.
	INT64	The INT64 data type is capable of storing a wide range of integer values, both positive and negative, within the constraints of a 64-bit binary representation. The range of values for a 64-bit integer is extensive, making it suitable for applications where very large or very small integer values need to be handled.
DATETIME	DATETIME (Default)	DATETIME data types store information about both the date (day, month, and year) and time (hours, minutes, seconds, and sometimes fractions of a second).
	DATE	Date fields store date values, including options for time and time zone settings.
	TIME	Time fields store time values without date information.

Data Type	Data Format	Description
	DURATION	Duration fields are used to represent a specific length of time, typically measured in terms of hours, minutes, seconds, and sometimes milliseconds. It allows you to work with time intervals or durations, as opposed to specific points in time, which are handled by date and time data types.

> **Note**
>
> - You can add up to 100 codes in a code list field. For existing static code lists having more than 100 codes, further code addition is restricted. However, other modifications such as activation / deactivation of the codes are allowed.

10.1.2 Field Attributes

Field attributes, also known as column attributes or field properties, are characteristics or settings associated with a database field (column) that define various aspects of how data is stored, displayed, and processed within a database. These attributes help ensure data integrity, enforce constraints, and control how the data is used.

Available Field Attributes

Field Attribute	Description
creatable	Createable fields indicate whether a field can be populated with data when inserting a new record (i.e., during the creation of a new record in a database table). If a field is createable, you can provide a value for it when adding a new record to the database. If it's not createable, the field must remain empty or be set to a default value during insertion.
updatable	Updateable fields determine whether a field's value can be modified after the record has been created. If a field is updateable, you can change its value when updating existing records. Fields marked as non-updateable typically remain unchanged after the record's creation, but this behavior can be controlled using field attributes.

Field Attribute	Description
nullable	A field is nullable if it can contain null values, meaning that it's not required to have a value in every record. If a field is nullable, you can leave it empty (null) when creating or updating a record. If a field is not nullable, it must always have a valid value, and it can't be null. Non-nullable fields are typically used for data that's required and must always have a value.
required	Required fields are those that must have a value in every record, and null values are not allowed. When defining a field as required, you must ensure that it is populated with a valid value whenever a new record is created or an existing record is updated. These fields are essential for data integrity and accuracy.
hidden	Hidden fields are not visible to the user by default, but they may contain data that is used for various purposes, such as calculations or internal processing. Hidden fields are often used in forms or data structures where the data is needed for processing but doesn't need to be displayed to the user.
multi value	Multi-value fields can hold more than one value at a time, as opposed to single-value fields that store only one value per record. If a field is marked as multi-value, you can assign multiple values to it when creating or updating a record. For example, a field for product categories could store both **Electronics** and **Accessories** at the same time.

Note

- You cannot change the `searchable` property for the `OBJECT`, `ARRAY`, `INTEGER`, `NUMBER`, and `DATETIME` datatypes.
- You cannot change the `filterable` and `sortable` properties for the `OBJECT`, `ARRAY`, `INTEGER`, and `NUMBER` datatypes.
- The `required` and `nullable` properties are not applicable for the `BOOLEAN` datatype.
- `OBJECT` `dataType` can not have `dataFormat` as `DEFAULT` or not defined. `dataFormat` `AMOUNT` or `QUANTITY` or `TEXT` are only supported for custom/extension attributes.
- You cannot update `dataType` and `dataFormat`.

Procedure

1. Navigate to *User Menu* > *System Settings* > *All Settings* > *Extensibility* > *Extensibility Administration* .
2. In the *Field Attributes* tab, set the createable, updateable, and required properties by turning on the switch next to them.
3. Optional: Mark the fields as multi-value by selecting the checkbox next to the fields.

10.2 Create Extension Fields in SAP S/4HANA for Utilities

Extend a standard function module in SAP S/4HANA by adding extension fields to the API request and exposing those fields through the service definition so they are available externally.

Procedure

1. Identify the relevant function module / RFC used by the utilities integration.
2. Use the transaction `SE11` to review the parameter structure that contains the output fields. This structure contains standard fields sent by the API.
3. Extend this structure with custom fields. For this, In the menu bar choose *Goto* and *Append Structure*.
4. Create a new append structure. This append structure will hold your custom fields.
5. Define all required custom fields inside the append structure.
6. Locate the enhancement points provided in the function module code.

   These are predefined hooks where customer logic can be implemented. They enable you to insert custom logic without modifying SAP standard code. You may also use implicit enhancements (pre-exit, post-exit) if available.
7. Create an enhancement implementation using the provided enhancement points.
8. Open the corresponding SOAP service proxy and go to the service definition in the edit mode.
9. Perform a syntax check, if extended fields were added, there will be a prompt to regenerate the service metadata.
10. Choose *Regenerate* to refresh the service definition. This ensures the custom fields are included in the external API (SOAP payload).
11. Activate the updated service definition.

    Without regenerating the service definition, the custom fields added to the RFC structures will not be visible outside the SAP S/4HANA system.

    > → Tip
    >
    > To access the list of enhancement spots along with the enhancement points and sections, run the program (report) `ISU_C4C_V2_FM_ENHANCEMENTS` using the `se38` transaction in SAP S/4HANA.
    >
    > Select the package as `ISU_C4C_INT`.

10.3 Integrate Using Cloud Integration (CI)

After creating custom fields in SAP Service Cloud Version 2 and SAP S/4HANA, you must integrate the OpenAPI schema generated from SAP Service Cloud V2 and the WSDL file from SAP S/4HANA.

The platform integration packages include both synchronous integration flows (transactional data) and asynchronous integration flows (master data).The types of integration flows are as follows:

- **Asynchronous flow**: Initiating from the S4H system in XML format, the integration flow includes mapping that converts XML data into JSON format before sending it to the SAP Service Cloud Version 2 system.
- **Synchronous flow**: These consist of two types: **Get calls** and **Post calls**. Get calls retrieve transactional data from S4H, while Post calls retrieve data and create, update, or delete data in the S4H system.
    - **Get Integration Flow**: Originating from the SAP Service Cloud Version 2 system via HTTP query, this flow uses two Groovy scripts: one processes HTTP query parameters to structure JSON in message headers, and the other converts JSON data into custom XML format stored in the message body. The XML request is then sent to the S4 system, which responds with XML. Response mapping converts the XML response into JSON format.
    - **Post Integration Flow**: Starting from the SAP Service Cloud Version 2 system in JSON format, this flow undergoes request mapping before sending the XML request to the S4H system. The S4H system responds with XML, and response mapping converts the XML response into JSON format before sending it back to the SAP Service Cloud Version 2 system.

All integration flows use either message mapping, Groovy scripts, or a combination of both to map data formats between systems, converting between XML and JSON as needed.

10.3.1 Custom Extensions

Based on the customers requirements, some enhancements are needed at the mapping level to allow for the transfer of additional information between SAP Service Cloud Version 2 and S4H systems, which are not included in the standard mapping content but are specific to the customers implementation.

> ⓘ Note
>
> One way of handling this is editing the standard integration flows. It is important to note that editing the standard integration flows will result in them no longer receiving updates from SAP. Therefore, it is recommend enabling the extensibility feature which are provided for all the utilities standard integration flows.

Custom Exists

The integration flows are designed with custom exit points, allowing users or customers to enhance SAP delivered content without modifying the standard integration flow. This ensures that users or customers are always notified of future patches and enhancements for those artifacts.

Each integration flow offers parameters that determine the need for a custom extension: Custom Exit for Asynchronous, Request Exit, and Response Exit for Synchronous. If the parameter value is set to **true**, the

source payload is saved to a message header property and makes a call to the local integration flow via the process direct adapter. If the parameter value is set to **false**, the default route is taken, and the standard mapping is applied without further extension.

In the local integration flow, customers or users can specify their requirements without modifying any standard integration flows.

> ⓘ Note
>
> Both standard and custom integration flows are deployed on the same tenant and processed together, communicating with each other using the **Process Direct Adapter**.

10.3.1.1 Asynchronous Replication

Let us consider **Replicate Utilities Premise Connection Object from SAP S/4HANA** as an example to understand the flow in detail.

This integration flow replicates Premise and Connection Object master data from SAP S/4HANA to SAP Service Cloud Version 2.

The messages are sent via SOAP (XML) format from SAP S/4HANA to SAP Cloud Integration, where they are converted to JSON format and delivered to the SAP Service Cloud Version 2 system. The integration flow includes a parameter (Custom Exit) that determines if a custom extension is needed. If the parameter is set to **true**, the source payload is saved to a message header property and an exit flow call is made via the **Process Direct** adapter.

The address of the process direct adapter is externalized and can be configured during the integration flow setup. Customers must ensure that the address of the process direct receiver adapter matches the one used in the process direct sender adapter of the local integration flow.

> ⓘ Note
>
> Due to differing data formats, they cannot be combined into a multi mapping message (which only supports combining messages of a single data format) in the local integration flow. Therefore, a Groovy script is used to enrich the message to meet customer-specific requirements

The sample Groovy code snippet is as below:

> {·} Sample Code

```
import com.sap.gateway.ip.core.customdev.util.Message
import groovy.json.JsonOutput
import groovy.json.JsonSlurper
import groovy.util.XmlSlurper
def Message processData(Message message) {
  def body = message.getBody(java.io.Reader)
  def json = new JsonSlurper().parse(body)
  def xmlString = message.getHeader("sourcePayload",java.io.Reader)
  def xml = new XmlSlurper().parse(xmlString)
  json.messageRequests.eachWithIndex { request, index ->
    // Extracting the first field
    def previousName1 =
xml.PremiseConnObjReplicateRequestMessage[index].PremiseConnectionObject.Premi
seType.text()
```

```
    // Extracting the second field
    def previousName2 =
xml.PremiseConnObjReplicateRequestMessage[index].PremiseConnectionObject.Conne
ctionObjectId.text()
        // Extracting the third field
    //def previousName3 =
xml.PremiseConnObjReplicateRequestMessage[index].PremiseConnectionObject.comme
nt2.text()
    // Adding the first field to the extensions array
    request.body.extensions = ["Ext_PremiseType": previousName1]
    // Adding the second field to the extensions array
    request.body.extensions << ["Ext_connectionObjectID": previousName2]
    // Adding the third field to the extensions array
    //request.body.extensions << ["Demo_TECH": previousName2]
  }
  def modifiedJson = JsonOutput.toJson(json)
  message.setBody(modifiedJson)
  return message
}
```

The following codes are executed by the script:

1. **Read the JSON and XML Data**: The script retrieves JSON data from the message body and XML data from the message header.
2. **Extract Information from XML**: It extracts specific information from the XML data.
3. **Add Information to JSON**: The extracted information is then added to each JSON request as additional details.
4. **Update the Message**: The script converts the updated JSON data back to a string format.
5. **Return the Updated Message**: Finally, it sets the updated JSON as the new message body and returns the message.

The modified/enriched JSON is sent to main integration flow which in turn is delivered to target system.

10.3.1.2 Synchronous Replication

Synchronous extensibility allows users to customize integration flows to meet specific business requirements, such as adding new fields or custom logic to the integration process.

To illustrate this, lets examine the **Create Bank Account for Business Partner Integration Flow**, which handles sending requests from SAP Service Cloud Version 2 to SAP S/4HANA, which then responds with a message back toSAP Service Cloud Version 2.

Messages are initially sent in JSON format from SAP Service Cloud Version 2 to SAP Service Cloud Version 2, where they undergo request mapping to convert them into SOAP (XML) format before being delivered back to the SAP S/4HANA system. The SAP S/4HANA system responds in SOAP (XML) format, and response mapping converts the XML data to JSON format and sends it to theSAP Service Cloud Version 2 system.

The integration flow includes parameters i.e Request Exit and Response Exit, which determines if a custom extension is needed or not. If the parameter is set to true, the source payload is saved to a message header property as **CNSRequestPayload**. A Groovy script in the request mapping is executed to convert the JSON format to XML format, and then the Request Exit flow is called via the process direct adapter.

Request Exit

In the **Request Exit** integration flow, a Groovy script takes the data stored in the message header property of the main integration flow, extracts specific fields from the header, adds custom fields, and forms the XML body payload. It then returns the updated message, now containing both standard fields and customer extension fields, which is sent back to the main integration flow.

The sample Groovy code snippet is as below:

Sample Code

```
import com.sap.gateway.ip.core.customdev.util.Message
import groovy.xml.XmlUtil
import groovy.util.XmlSlurper
import groovy.json.JsonSlurper
def Message processData(Message message) {
 // Get header value
 def header = message.getHeaders().get("CNSRequestPayload")
 def jsonSlurper = new JsonSlurper()
 def headerValue = jsonSlurper.parseText(header as String)
 // Extract required fields from header
 def BusinessPartnerId = headerValue.extensions.ZbusinessPartnerId
 //def premiseId = headerValue.extensions.ZpremiseId
 // Get existing body payload
 def bodyPayload = message.getBody(String) ?: ''
 def xml = new XmlSlurper().parseText(bodyPayload)
 // Add extracted fields to the XML body
 xml.appendNode {
 IvBusinessPartner(BusinessPartnerId)
 // IvLanguage(premiseId)
 }
 // Convert updated XML back to string
 def updatedPayload = XmlUtil.serialize(xml)
 // Set updated payload to message body
 message.setBody(updatedPayload)
 return message
}
```

The following codes are executed by the script:

1. **Retrieve Header Data**: The script retrieves the value of a specific header from the message, which contains JSON data.
2. **Extract Information from JSON**: It passes this JSON data to extract specific fields, such as the Business Partner ID.
3. **Read Existing XML Body**: It reads and passes the existing XML data from the message body.
4. **Update XML with New Data**: It adds the extracted information from the JSON into the XML data as new elements.
5. **Update and Return Message**: The script converts the updated XML back to a string format, sets this updated XML as the new message body, and returns the message.

Response Exit

The request XML data is sent to the SAP S/4HANA system via a SOAP adapter, and the system responds with XML containing both standard and custom fields. This data is saved as **S4HResponsePayload** in the main

integration flow. The standard response message mapping is then executed, but the output does not include the custom fields. After the response message mapping, a response Exit call is made, which triggers **Response Exit Integration Flow**.

The Response Exit integration flow uses a Groovy script to handle the message by extracting XML data from the header **S4HResponsePayload**, adding specific XML content as an extension to the JSON payload in the message body, and updating the message body with the modified JSON. It then returns the updated message, which now includes both standard and custom fields, to the main integration flow. The main integration flow subsequently sends the response back to SAP Service Cloud Version 2.

The sample Groovy code snippet is as below:

Sample Code

```
import com.sap.gateway.ip.core.customdev.util.Message
import groovy.json.JsonOutput
import groovy.json.JsonSlurper
import groovy.util.XmlSlurper
def Message processData(Message message) {
    def body = message.getBody(java.io.Reader)
    def json = new JsonSlurper().parse(body)
    def xmlString = message.getHeader("S4HResponsePayload", java.io.Reader)
    def xml = new XmlSlurper().parse(xmlString)
    // Extracting the first field
    def previousName1 = xml.EsBankDetail.BankName.text()
    // Assuming json is a single record and not a list
    // Adding the extensions segment to the json
    json.value.extensions = [
        zBankName: previousName1.toString()
        // Add more fields as necessary
    ]
    // Convert the modified JSON object back to a JSON string
    def modifiedJson = JsonOutput.toJson(json)
    message.setBody(modifiedJson)
    return message
}
```

The following codes are executed by the script:

1. **Read the JSON Body**: The script retrieves and reads the JSON data from the message body.
2. **Get XML Data from Header**: It extracts XML data from a specific header in the message.
3. **Extract Data from XML**: The script extracts specific information (e.g., bank name) from the XML.
4. **Update JSON with Extracted Data**: It adds the extracted information as an extension to the JSON data.
5. **Update and Return Message**: The script converts the updated JSON data back to a string format, sets it as the new message body, and returns the message.

11 Appendix

Check Endpoint Name in SAP Cloud Integration (SCI)

1. Connect to the Tenant Management node of the SAP Cloud Integration with https://<tenant management node>/itspaces.
2. At the top left, choose the *Monitor / Operations* view icon.
3. Under *Manage Integration Content*, choose *All*.
4. Select the needed *Integration Flow* from the list and copy the URL/path under *Endpoints*.

Ticket Component

In case that you have any issues, please raise a case via the SAP Support Portal via https://support.sap.com/en/index.html

Component	Comment
CA-GTF-INT-CFC	For issues relevant to documentation
CEC-CRM-UTIL	For other technical issues

12 Utilities Sales Process Integration with SAP S/4HANA

This document describes the procedure to manually configure the integration between SAP S/4HANA and SAP Service Cloud Version 2 for Utilities using SAP Cloud Integration (SCI).

If you have any queries or feedback about the Integration Package SAP Sales Cloud Version 2 Integration for Utilities Sales Processes with SAP S/4HANA mentioned in this document, please create a ticket using the component *CEC-CRM-UTIL*.

12.1 Prerequisites

Prerequisites for integrating SAP S/4HANA with SAP Service Cloud Version 2.

You've access to SAP Service Cloud Version 2 system which includes:

- Initial user and access information for your SAP Service Cloud Version 2 tenant provided via email by Cloud Management Service.

You've access to SAP Cloud Integration system. This includes:

- Integration developer persona assigned to your user in your SAP Cloud Integration tenant to import and deploy integration flows on SAP Cloud Integration.
- Initial user and access information for your SAP Cloud Integration tenant.
- SAP Cloud Integration Client Certificate.

Required Information

While performing the activities described here, you must enter or provide system-specific information. Before you start the integration process, it's recommended that you have the information listed. The information ensures a smooth and efficient integration of the SAP S/4HANA and SAP Service Cloud Version 2.

- SAP Service Cloud Version 2
- SAP S/4HANA system
- SAP Cloud Integration
- SAP Cloud Connector

12.2 Configuration in SAP Service Cloud Version 2

12.2.1 Create Communication System

A communication system represents an external system that is used for application integration. Communication systems can be, for example, an external time recording or master data system.

Context

On the *Communication Systems* page, you can filter the communication systems by status and authentication type. You can create and edit communication systems to exchange data across systems electronically.

Procedure

1. Log in to the system as an administrator.
2. Navigate to user profile on the right-top corner, and access the *System Settings* page.
3. Go to *All Settings* > *Integration* > *Communication Systems*.
4. Click the + (Add) icon on the *Communication Systems* page, enter the *Display ID* and *Description* for the communication system.

   > **Note**
   >
   > Display ID must be the Business System ID of your external system.

5. Click *+ Create* button in the *Inbound* tab, enter a password or upload a certificate file for authentication.
6. Click *Save and Activate* to create the inbound communication system.

   > **Note**
   >
   > By default the user is created while creating communication system. The communication user ID is same name as the communication system and the password is maintained in the inbound communication tab of the communication system.

7. Click *+ Create* button in the *Outbound* tab, choose a protocol, enter a host name, select an authentication method from the dropdown, and click *Save and Activate* to create the outbound communication system.

Results

You've now successfully created a communication system, which you can use for integrating applications.

Related Information

Create Communication Systems

12.2.2 Create Communication Configuration

Configure Utilities Sales Process communication between SAP Service Cloud Version 2 and the SAP S/4HANA you are integrating it with.

Procedure

1. Go to *User Menu* > *System Settings* > *All Settings* > *Integration* > *Communication Configuration*.
2. To integrate Service Cloud Utilities Master Data with SAP S/4HANA, select the following communication configurations:
   - Integrate Service Cloud Utilities with SAP S/4HANA – Utilities Sales

   You can also choose + *Create* to create a communication configuration for your requirements.
3. Choose *Copy*.
4. Go to the *Communication Configuration* page and click the copied communication configuration.
5. Edit *Communication System* and select the name of the communication system.

   Target message entity is the structure in which the target system expects the data to arrive. *Target Message Entity API Path* is the path to which the data is stored. Source entity is the structure in which the source sends the data to connector.

   The test icon gives the connection status.

   - The *Target Message Entity API Path* must match the respective iflow endpoints.
   - For all the modules except Integrate Service Cloud Utilities Master Data with SAP S/4HANA, the synchronous calls and services list are under HTTP data. For Integrate Service Cloud Utilities Master Data with SAP S/4HANA, the synchronous calls and services list are under Inbound Configuration.
6. In *HTTP Data*, use the search icon to select the configurations relevant to your scope.

12.3 Configuration in SAP S/4HANA

An overview of how to configure Utilities Sales Process integration with S/4HANA system.

12.3.1 Prerequisites

Prerequisites for configuring Utilities Sales Process integration with SAP S/4HANA.

Ensure the administrator has business user access before starting the configuration in SAP S/4HANA. Upgrade the software component *IS-UT 809 FPS0* and implement SAP Note 3770164 to enable all features of SAP Sales Cloud Version 2 Integration for Utilities Sales Processes with SAP S/4HANA.

12.3.2 Maintain Trust between SAP Cloud Integration and SAP S/4HANA

Trust relationships enable you to establish secure connections between known servers whose identity can be confirmed by a signed certificate.

Context

You need to download the certificate of SAP Cloud Integration tenant via a web browser and import it to SAP S/4HANA. Google Chrome is used as the default browser.

Procedure

1. Access SAP Cloud Integration using the URL https://%3Ctenant%20management%20node%3E/itspaces
2. In the browser, choose the lock icon and then *Certificate* (Valid) of your secure connection.
3. In the *Certificate* dialog box, go to the *Details* tab.
4. Choose *Copy to File*.
5. In the *Certificate Export Wizard* dialog box, choose *Next*.
6. Choose *Base-64 encoded X.509 (.CER)* and choose *Next*.
   Certificates can be exported in a variety of file formats.
7. Specify the name of the file and save it to the desktop by default. Choose *Next*.
8. Choose *Finish*.

Results

A message tells you that the export was successful.

Upload SAP Cloud Integration Certificate to S/4HANA

Context

Uploading SAP Cloud Integration certificates to SAP S/4HANA is a crucial step in establishing a secure communication setup between the two systems. Certificates are used for authentication, ensuring that the communication between SAP Cloud Integration and SAP S/4HANA is secure and trusted.

Procedure

1. Log on to your SAP S/4HANAsystem as an administrator (**SAP_BR_ADMINISTRATOR**).
2. Open *Maintain Certificate Trust List*.
3. Next to the *Certificate Trusts* table, choose + (Add icon).
4. In the *Upload Certificate* dialog box, browse for the SAP Cloud Integration certificate that you've downloaded and choose *Upload*.

Results

A system message informs you that the certificate has been uploaded.

12.3.3 SAP S/4HANA Web Service Configuration (SOA Management)

This feature is intended for technical system administrators who configure inter-system connections and manage security settings in the system landscape.

To use web services with Web Services Reliable Messaging, you must have the web service runtime configured.

The configuration of the web service runtime is client-specific and must be performed in each productive client.

User Action	Entry Value
Transaction Code	SOAManager

12.3.3.1 Utilities Sales Process Inbound Web Service Configuration

Activate the SAP S/4HANA inbound web service to set up the connection for communication from SAP Cloud Integration to SAP S/4HANA.

Context

You can activate the inbound web services listed below for communication from SAP Cloud Integration to the SAP S/4HANA system by enabling the service definitions in SAP S/4HANA:

- SALESORDERBULKREQUEST_IN
- CRMS4_IU_CM_PROCESS_SD_MON

Procedure

1. Go to the *SOA Management* (SOAMANAGER) transaction in your SAP S/4HANA system.
2. Choose *Simplified Web Service Configuration*.
3. Enter the web service definition name in the *Search Pattern* field and choose *Go*.
4. In the result list, select the Username/Password (Basic) check box for the web service definition.
5. Choose *Save*.

12.3.3.2 Utilities Sales Process OData V4 Service Configuration

Context

To ensure the Utilities Sales Process functions properly, publish multiple OData V4 services in the S/4HANA system.

Publish the following services:

- API_UTILITIESPRODUCT
- API_UTILITIESSALESCONTRACT
- API_UTILSSALESCONTRACTPROCESS

Procedure

1. Open transaction **/IWFND/V4_ADMIN** in your SAP S/4HANAsystem.
2. Select *Publish Service Groups* at the top of the screen.
3. Enter your system alias.
4. Search for the unpublished service group in the *Not Published Service Groups*.
5. Select the service group you want to publish.
6. Select *Publish Service Group*.
7. Enter a description for the service group.
8. Press *Get Service Group* to complete the publishing process.

Results

After you complete the publishing process, the service group appears in the list of published services in transaction **/IWFND/V4_ADMIN**.

12.4 Configuration in SAP Cloud Integration

12.4.1 Copy Integration Package

The prepackaged Utilities Sales integration package SAP Sales Cloud Version 2 Integration for Utilities Sales Processes with SAP S/4HANA contains integration flows that need to be copied and configured.

Procedure

1. Log in to the SAP Cloud Integration tenant.
2. From the *Menu*, select *Discover*.
3. Choose *Integration*.
   All the SAP delivered integrations packages are listed.
4. Search and select the utilities *SAP Sales Cloud Version 2 Integration for Utilities Sales Processes with SAP S/4HANA* package.
5. Choose *Copy*.

12.4.2 Deploy Security Credentials

The communication between SAP Cloud Integration and SAP Service Cloud Version 2 is established via basic authentication over HTTPS. User credential needs to be deployed in the SAP Cloud Integration tenant, which is used for sending messages to SAP Service Cloud Version 2.

Procedure

1. Log in to the SAP Cloud Integration tenant.
2. From the *Menu*, select *Monitor*.
3. Under *Manage Security*, select *Security Material*.
4. From the *Create* dropdown, select *User Credentials*.
5. In *Create User Credentials*, enter the required details.

   Use the credential maintained in *SAP Service Cloud Version 2* > *Settings* > *Integration* > *Communication Systems* > *Inbound*.
6. Choose *Deploy* to deploy the user credential artifact.
7. For basic authentication of inbound messages in SAP S/4HANA, repeat the steps for the user authorized to execute messages in the SAP S4H system.

12.4.3 Configure and Deploy Utilities Sales Process Integration Flows Individually

Integration flow configuration and deployment for connecting SAP Sales Cloud Version 2 with SAP S/4HANA in Utilities Sales Processes. Use it to establish secure communication between cloud and on-premise systems by configuring sender authorization, receiver endpoints, authentication credentials, and optional Cloud Connector settings.

1. Log in to the SAP Cloud Integration tenant.
2. To configure the integration flows, select *Design* from the menu.
3. Select the copied integration package *SAP Sales Cloud Version 2 Integration for Utilities Sales Processes with SAP S/4HANA*.
   All integration flows are displayed on the *Artifacts* tab.
4. Select all required integration flows or, select *Select All*.
5. From the *Actions* menu, select *Configure*.
6. Configure the sender system (SAP Service Cloud Version 2) and receiver system (SAP S/4HANA) details:
   1. In the *Sender* tab, select the *Authorization and User Role* options.
   2. Configure the receiver address in the *Receiver* tab.

Configure Receiver Address

The receiver address represents the endpoint URL of the receiving SAP S/4HANA system.

You can maintain the receiver address using two approaches:

The URL must start with the protocol, host, and port of the SAP S/4HANA system. By default, the system maintains the S/4HANA host, port, and client as integration flow properties. To determine the inbound endpoint URL for each SAP S/4HANA service, navigate to transaction *SOAMANAGER* in the SAP S/4HANA system.

Maintain Receiver Address Using Properties (Recommended)

1. Open the integration flow and navigate to the *More* tab.
2. Select the *Receiver* tab.
3. Maintain the receiver address using the properties for host, port, and client.

This approach is recommended because the SAP client may already be included as a query parameter in the endpoint URL. In addition, you must maintain the receiver address consistently for both:

- The S4H receiver
- The S4HCSRF receiver configuration

Using properties ensures centralized and consistent configuration.

Maintain Receiver Address Using Direct URL Replacement

1. Open the integration flow and navigate to the *More* tab.
2. Select the *Receiver* tab.
3. Replace the placeholders *${property.LocalS4HostandPort}* and *${property.LocalS4Client}* with the actual SAP S/4HANA host, port, and client values. For example, http://ldcabc.wdf.sap.corp:56007/sap/bc/srt/pm/sap/salesorderbulkrequest_in?sap-client=001&MessageId=${property.uuidMessageId}

> → Recommendation
>
> SAP recommends maintaining the receiver using properties, as it avoids duplication of client information in the URL and simplifies maintenance across multiple receiver configurations.
>
> 1. From the *Proxy Type* dropdown, select *On-Premise* (for connectivity via SAP Cloud Connector) or *Internet* (for connectivity via reverse proxy-based infrastructure).
> 2. If you use a Location ID, update it. If no Location ID is required, leave this parameter blank. If you use a Location ID, ensure that you maintain it consistently across all relevant receiver configurations. You can maintain the Location ID using two options:
>
>   Update Location ID in Receiver Tab
>
>   1. Navigate to the *More* tab.
>   2. Update the parameter *P_S4_LocationID* with the required value.
>
>   Configure Authentication Credentials
>
>   1. Navigate to the *Receiver* tab.
>   2. Update the Location ID by maintaining the property *property${property.LocalLocationID}*.

> **Note**
>
> You can connect multiple Cloud Connectors to a subaccount if each has a unique Location ID. The Location ID identifies which Cloud Connector the system uses to establish the connection. By default, the Location ID is an empty string, which indicates the Cloud Connector connected without a specific Location ID.
>
> 1. From the *Authentication* dropdown, select *Basic*.
> 2. In the *Credential Name* field, enter the name of the credential deployed in security artifacts.
> 3. Repeat this procedure for all integration flows that have SAP Service Cloud Version 2 as the sender and SAP S/4HANA as the receiver.

12.4.4 Configure Utilities Sales Master Data Replication

Integration flow configuration for replicating S/4HANA utilities data to SAP Sales and Service Cloud systems. Use it to establish secure connections between sender and receiver systems by configuring authentication, host settings, and deployment credentials for premise connections and contract accounts.

Procedure

1. Select the integration flow *Replicate Utilities Premise Connection Object from S4HANA* that you want to configure.
2. Select the details for the sender system (S4H), and the receiver systems (SAP Sales Cloud and SAP Service Cloud Version 2 and S4H).

   In the *Sender* tab, select the *Authorization* and the *User Role* options.
3. Choose the *Receiver* tab and create the following entries:
   a. Host: SCV2 tenant's URL (without https://)
   b. Port: 443
4. Choose proxy type as *Internet*.
5. Choose *Allow Chunking*.
6. Select *Basic Authentication* from dropdown.
7. Provide the credential name which is deployed under security artifacts.
8. Choose *Save*.
9. Choose *Deploy*.

   Repeat the same procedure for the integration flow Replicate Utilities Contract Account from S/4HANA.

12.5 Configuration in SAP Cloud Connector

SAP Cloud connector allows SAP BTP applications to securely access remote services that run on the Internet or SAP S/4HANA.

Procedure

1. Log on to the SCC administration UI.
2. Select the relevant sub account.
3. Choose *Cloud to SAP S/4HANA* under the sub account menu.
4. Under *Access Control* tab select + (Add) button in *Mapping Virtual To Internet System* section.

Field Name	Entry Value
Back-end Type	ABAP System
Protocol	HTTPS
Internal Host	ldciqvp.devsys.net.sap (example)
Internal Port	44300
Virtual Host	qvp (example)
Virtual Port	443
Principal Type	None
Host in Request Header	Use Virtual Host
Description	(Not mandatory)
Check Internal Host	True

5. After each step select the *Next* button and then *Finish*.
6. Select the newly created entry.
7. On the Resources of qvp:443 (example) section select + (Add) button.

Field Name	Entry Value
URL Path	/sap/bc/srt/
Activate	True

8. Save the entry.
9. To check if the configuration was done correctly follow these steps:
   a. Open SAP BTP Cockpit.
   b. Navigate to the *Connectivity* tab.
   c. Select *Cloud Connectors*.
   d. Check if the new entry is available under *Exposed Back-end System* section.

12.6 Replication of Master Data Initial Load

In the following sections, we describe the replication of contract account, utilities technical master data (Connection object and premise), and business partner.

Procedure

1. Load contract account replication
2. Load technical object replication.
3. Load business partner and business partner replication.

> **Note**
>
> - Start loading data in small chunks, check if they are loaded successfully, and then later increase the size of the data.
> - After loading the data, go to *SAP Service Cloud Version 2* > *Integration* > *Message Monitoring* and check the status of the operation.

Related Information

Technical Master Data Replication [page 28]
This data includes connection objects and premises replicated to the SAP Service Cloud Version 2.

Data Replication for Point of Delivery [page 30]
This data includes connection objects and premises replicated to the SAP Service Cloud Version 2.

Load Business Partner and Business Partner Relationships [page 33]
For business partners you must use data replication framework.

12.6.1 Setting Up SAP S/4HANA for Contract Account Replication

Set up SAP S/4HANA to replicate contract accounts to SAP Service Cloud Version 2.

12.6.2 Technical Master Data Replication

This data includes connection objects and premises replicated to the SAP Service Cloud Version 2.

The old procedure has been deprecated, so please follow the latest setup instructions for configuration.

12.6.3 Data Replication for Point of Delivery

This data includes connection objects and premises replicated to the SAP Service Cloud Version 2.

12.6.3.1 Setting Up SAP S/4HANA Utilities for POD Replication

Set up SAP S/4HANA Utilities to replicate Points of Delivery (POD) to SAP Service Cloud Version 2.

Context

Create the POD outbound service in the SOA Manager tool for the data exchange from SAP S/4HANA Utilities to SAP Service Cloud Version 2.

Procedure

1. Use transaction **`SOAMANAGER`** from the SAP S/4HANA system.
2. Under the *Service Administration* tab, choose *Web Service Configuration*.
3. Search for the object name **`CO_ISU_C4C_V2_POD_REPL`** or **`UtilitiesPointOfDeliveryReplicateRequest_Out`**.
4. Choose the internal name of the object.
5. Choose *Create* and then select *Manual Configuration*.
6. Configure the logical port name as follows and then choose *Next*.
   a. In the *Logical Port Name* field, enter the logical port name of the system to which you're connecting.
   b. Select the *Logical Port as Default* checkbox.
   c. Enter a description such as **`Utilities POD Replication`**.
7. Configure the consumer security with one of the following forms of authentication and then choose *Next*.
   - **Basic Authentication**
     Enter the user ID and password that you use to connect to your Cloud Integration tenant.

     > ⓘ Note
     >
     > By default, the *User ID / Password* radio button is selected.

   - **Certificate-Based Authentication**
     1. Select the *X.509 SSL Client Certificate* radio button.
     2. Choose the value help of the *SSL Client PSE of transaction STRUST* field.
     3. Choose the PSE in which the client certificate that was issued by Certification Authority (CA) and supported by Cloud Integration is stored.

8. Configure the HTTP settings as follows and then choose *Next*.
    a. In the *URL* field, enter the runtime URL from Cloud Integration and append the following string to it: /cxf/Utilities/PremiseAndConnectionObjectReplication.
    b. Enter a name such as Proxy for the proxy host.
    c. Enter a port number such as 8080 for the proxy host.
9. Configure the SOA protocol as follows and then choose *Finish*.
    a. In the *RM Protocol* dropdown menu, select *SAP RM*.
    b. In the *Message ID Protocol*, select *SAP Message ID*.
    c. In the *Data transfer scope* dropdown menu, select *Basic Data Transfer*.

12.6.3.2 Integration in SAP S/4HANA

Understand the steps required to configure the Data Replication Framework (DRF) for Point of Delivery (POD) replication.

Procedure

1. Access DRF customizing using the transaction `DRFIMG`.
2. Navigate to *Data Replication* > *Define Custom Settings for Data Replication* > *Define Technical Settings* > *Define Technical Settings for Business Systems*.
3. Define the business system.
    a. Add a new entry for the business system **SAP Marketing Cloud** and save your changes.
    b. Select the business system and double-click *Define Bus. Systems, BOs* in the dialog structure.
4. Define the business object.
    a. Create a new entry with the *BO Type* as **`ISU_POD`** and *System Filter* as **`Selected`**, and save it.
    b. Select the business object type and double-click Define Bus. *Systems, BOs, Communication Channel*.
    c. From the *Communication Channel* dropdown, select *Replication via Services* and save.
5. Define the replication model.
    a. Navigate to *Data Replication* > *Define Custom Settings for Data Replication* > *Define Replication Models*.
    b. Create a new replication model with the following values:

Field	Value
Replication Model	Enter a suitable name.
Description	Enter a meaningful description
Log Days	50 (modify as required)

6. Assign outbound implementation.
    a. Select the replication model and double-click *Assign Outbound Implementation*.
    b. Create an entry for the *Outbound Implementation* as ISU_POD, and set the *Sequence* as 1.

7. Assign the target system.
    a. Select the outbound implementation and double-click *Assign Target Systems for Repl. Model / Outb. Impl.*
    b. Add a new entry using the business system created in step 3 and save.
8. Maintain outbound parameters.
    a. Select the business system and double-click *Assign Outbound Parameter*.
    b. Create a new entry with the *Output Parameter* as PACK_SIZE_BULK and the value as 20, or as required.
9. To activate the replication model, double-click *Define Replication Model*, select the replication model and choose *Activate*.
10. To enable delta replication, you must maintain the required workflow configuration.
    a. Go to transaction `SWE2`.
    b. Maintain the workflow event linkage for the `ISUPOD` business object.
    c. Ensure the relevant workflow for the event class CL_ISU_C4C_V2_POD_DRF_OUT is correctly configured and active for delta replication processing.

12.6.3.3 Maintain IMG Documentation

Procedure

1. Enhance the IMG structure.
    a. Go to transaction `S_IMG_EXTENSION` (Edit IMG Structure).
    b. Enter `SIMG_ISU` as the IMG Structure.
    c. Enter `ISU_C4C_S4H_UTI` as the Enhancement ID.
    d. Choose *Enhance Structure*.
2. Create the IMG node.
    a. Insert a subnode under *Cloud for Utilities*.
    b. Enter the document name and attach the document in the *Document* tab.
3. Fill the required details in the *Attributes* and *Customizing Activity* tabs.
4. In the *Document* tab, enter the following information: Using this configuration step, you can activate Point of Delivery replication across all services for the Utilities solution in SAP Service Cloud Version 2. This configuration enables the required web services to replicate Point of Delivery data from SAP S/4HANA Utilities to the solution
5. Save the main IMG node.

    You can use the following BAdI enhacements:

    - `MDG_ID_MAPPING_API` to modify the local system determination during Data Replication Framework processing.
    - `ISU_C4C_V2_TECH_OBJS_REPL` to modify sender and receiver information during Point of Delivery replication.

- `ISU_C4C_V2_REPLICATION_CONF` to dynamically change the logical port used for outbound replication.

12.6.4 Load Business Partner and Business Partner Relationships

For business partners you must use data replication framework.

Procedure

1. Run the *DRFOUT* transaction code.
2. In *Replication Model*, select your replication model.
3. In *Outbound Implementation*, select *986_3*.
4. In *Replication Mode*, select *Manual*.
5. Click *Manual replication Filter Criteria*.
6. Select the required business partners and relations.

12.7 Principal Propagation

Principal propagation is a secure, Single Sign-On mechanism that forwards an end-user's identity (principal) from a cloud application to an on-premise back-end system without changing it.

Principal propagation is supported only for synchronous transactions (SAP Service Cloud V2 to SAP S/4HANA) and is not supported for asynchronous integrations such as contract account and premise replication.

Related Information

Principal Propagation [page 34]
Principal propagation is a secure, Single Sign-On mechanism that forwards an end-user's identity (principal) from a cloud application to an on-premise back-end system without changing it.

Configure Trust [page 35]
To set up principal propagation, trust configuration must be established between SAP Service Cloud V2 and the SAP BTP subaccount where the SAP CPI tenant resides.

Configure Principal Propagation [page 36]
Principal Propagation allows the identity of a logged-in user to be securely forwarded through multiple systems without using technical users. The objective of this scenario is to propagate the logged-in SAP Service Cloud V2 user identity to SAP S/4HANA without using a technical user for business authorization.

12.7.1 Configure Trust

To set up principal propagation, trust configuration must be established between SAP Service Cloud V2 and the SAP BTP subaccount where the SAP CPI tenant resides.

Context

Based on internal testing, this method has been verified to work as expected. You may choose alternative methods depending on your project requirements.

Procedure

1. Create a key-pair using OpenSSL (self-signed certificate is also accepted).
2. Prepare an `IdPSSODescriptor` metadata XML file.
3. Upload the metadata XML as a Trusted Identity Provider in the SAP BTP subaccount.

Related Information

Set Up Trust

12.7.2 Configure Principal Propagation

Principal Propagation allows the identity of a logged-in user to be securely forwarded through multiple systems without using technical users. The objective of this scenario is to propagate the logged-in SAP Service Cloud V2 user identity to SAP S/4HANA without using a technical user for business authorization.

Context

In the following example setup, the principal of the outbound user is forwarded to SAP Cloud Connector, and from there to the back-end receiver system. We assume the following:

- In this example, the authentication option OAuth is used (using OAuth SAML 2.0 Bearer Assertion) for outbound communication from the sender to SAP BTP.
- SAP S/4HANA is the receiver system.
  The receiver system is connected to SAP BTP through SAP Cloud Connector.

> ⚠ Caution
>
> Using SAP Cloud Connector is mandatory when configuring principal propagation.

- To establish an outbound connection (SAP BTP to SAP Cloud Connector), an adapter that supports principal propagation is used (for example, the SOAP receiver adapter).

Procedure

1. Create a communication system in SAP Sales and Service Cloud V2, and use OAuth SAML 2.0 bearer assertion as the authentication method.
2. In the receiver channel of the integration flow, enable *Principal Propagation* as the *Authorization* method.
3. Prepare SAP Cloud Connector to support principal propagation with X.509 certificates (for the communication with the receiver system).

   You need a certificate chain with at least one intermediate certification authority. The intermediate certification authority signs a short-lived certificate, which is used for principal propagation. Use the user name (associated with the identity to be propagated) as the subject common name (subject CN) of this certificate.

   SAP Cloud Connector forwards the identity (to be propagated) in a short-living X.509 certificate in HTTP header SSL_CLIENT_CERT.
4. In SAP Cloud Connector, configure the trust relationship with the SAP BTP application.
5. Configure the receiver system. You need to do the following:
   a. Configure the receiver system to trust the certificate of the SAP Cloud Connector.
   b. Configure the Internet Communication Manager (ICM) to trust the system certificate for principal propagation.
   c. Map the short-living certificate (from SAP Cloud Connector) to the user (whose identity is being propagated).

Related Information

Setting Up Principal Propagation (Example Scenario)
OAuth 2.0 SAML Bearer Assertion Authentication

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 an 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.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.