Integration Guide | PUBLIC
SAP Integrated Business Planning for Supply Chain
Document Version: 1.0 – 2023-05-05

SAP Cloud Integration Guide
SAP Integrated Business Planning for Supply Chain 2305 and SAP Cloud Integration for Data Services

THE BEST RUN **SAP**

Content

1 Document History

> **i** Note
>
> Until three months after the publication of a new release of SAP Integrated Business Planning for Supply Chain, we publish regular documentation updates on the SAP Help Portal. If you use a local PDF copy or a paper printout of this document, make sure that you have the latest version. You can find it at http://help.sap.com/ibp2305.

The following table provides an overview of the most important changes.

Document History

Version	Date	Description
1.0	2023-05-05	Initial version for SAP Integrated Business Planning for Supply Chain 2305

2 Getting Started with Your Integration Project

SAP Cloud Integration for data services is the main integration tool used for all **time-series-based** tactical planning in the SAP Integrated Business Planning for Supply Chain (SAP IBP) solution. Using SAP Cloud Integration for data services, you can integrate master data, key figures, and time periods.

External master data and key figures that are typically relevant for applications focusing on order-based planning (OBP) can be integrated via SAP HANA smart data integration. However, the following data is integrated to OBP-relevant planning areas in SAP IBP using SAP Cloud Integration for data services:

- Transportation resources
- Lane resource assignments
- Product substitutions
- Data relevant for demand-driven replenishment (DDR) in OBP (transferred between planning areas within SAP IBP)

> → Recommendation
>
> Use planning areas based on flexible master data, since planning areas based on external master data are now outdated in order-based planning. In the past few releases, we have taken the following steps to prepare deprecation of planning areas based on external master data:
>
> - Since SAP IBP 2211, the SAP7 sample planning area has been hidden in the *Sample Model Entities* app in systems, where no existing planning area had been previously copied from the SAP7 sample. Instead, SAP7F, which is based on flexible master data, has become the default sample planning area recommended for order-based planning (OBP) projects.
> - As of SAP IBP 2305, new customers of order-based planning can no longer create (or activate) external master data types for OBP. The same holds true for existing SAP IBP systems: In systems, where external master data types haven't been used so far, they can no longer be created or activated.
>
> Customers who have been using a planning area based on external master data can still modify their existing model or copy from the SAP7 sample.
>
> In a future release, which is not yet defined, SAP will no longer support planning with external master data. We will announce this at least 12 months in advance.

For more information on the different data integration scenarios in SAP IBP, search for *Finding the Right Integration Tool* under *Data Integration Scenarios* on the Help Portal at http://help.sap.com/ibp.

> i Note
>
> SAP Cloud Integration for data services was previously called SAP Cloud Platform Integration for data services (CPI-DS), and before that, SAP HANA Cloud Platform, integration service for data services (HCI-DS). You might still find the old names at some places.

Related Information

2.1 Technical System Landscape

SAP Data Services Agent communicates directly with the SAP Integrated Business Planning for Supply Chain (SAP IBP) database.

Before setting up the integration of SAP IBP with your on-premise systems, please ensure that the following system landscape is in place:

- You have one or more on-premise systems in place, which serve as source systems for master data, transactional data, and time series data, for example, SAP S/4 HANA, SAP ERP, other ERP systems, SAP Advanced Planning and Optimization, or SAP Demand Signal Management. At the same time, you need target systems for the planning results in SAP Integrated Business Planning for Supply Chain.
  If you want to use a database table as source or target system, make sure your database license allows a direct access to the table.
- For SAP S/4HANA on-premise systems, we recommend to install and configure the SAP S/4HANA, Supply Chain Integration Add-On for SAP Integrated Business Planning to simplify the integration. For more information, see http://help.sap.com/ibp_s4hana_addon.
- For SAP ECC 6.0 systems (Enhancement Package 4 or higher) we recommend to install and configure the SAP ERP, Supply Chain Integration Add-On for SAP Integrated Business Planning to simplify the integration. For more information, see http://help.sap.com/ibp_erp_addon.
- You have instances of the following SAP delivered products running in the cloud:
  - SAP Integrated Business Planning for Supply Chain
  - SAP Cloud Integration for data services

Data Flow

The overall data flow looks as follows:
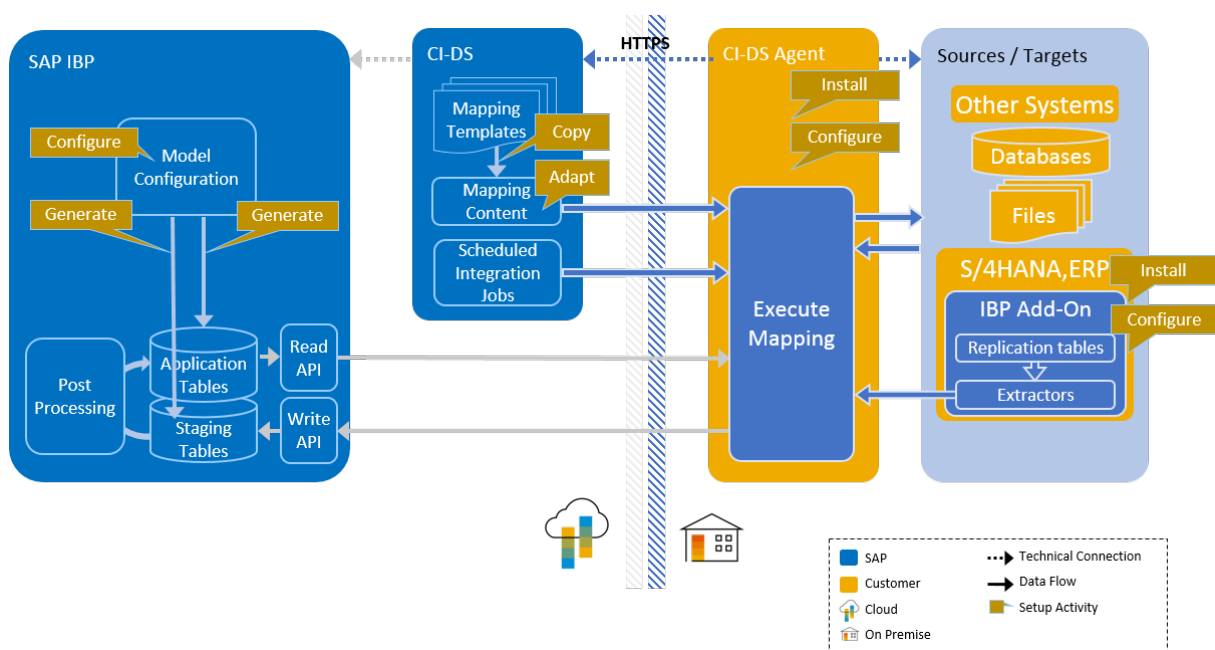
2.1.1 Setting Up Data Integration

This procedure involves configuration steps in multiple systems.

Set up data integration with SAP Cloud Integration for data services as follows:

1. In SAP IBP, activate your planning areas. Check the application log to make sure that application tables and staging tables have been generated successfully.
   When you configure your SAP IBP system, application tables and staging tables are generated. Application tables contain all the data needed for planning and also the planning results. Staging tables are used as a buffer for incoming data.

2. Install the SAP Data Services Agent in your on-premise landscape, either on a Windows or Linux server. For more information, see the SAP Data Services Agent Guide at https://help.sap.com/viewer/product/ SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES.

3. Configure the SAP Data Services Agent.
   Define all technical settings required for the communication with SAP Cloud Integration for data services in the cloud and with your on-premise systems, databases, and file shares.
   The technical connection between the SAP Data Services Agent and SAP Integrated Business Planning for Supply Chain is always triggered from the agent, so that you do not need to open your landscape for incoming connections from the SAP cloud. For more information, see the SAP Data Services Agent Guide at https://help.sap.com/viewer/product/SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES.

4. Define the communication arrangement in SAP IBP based on the SAP_COM_0593 communication scenario. The authentication method can be either certificate-based or basic authentication.
   For more details on setting up the communication arrangement, see Defining the Communication Arrangement [page 9].

5. Install and configure the SAP S/4HANA or SAP ERP, Supply Chain Integration Add-On for SAP Integrated Business Planning, if you decide to use it. For more information, see http://help.sap.com/ ibp_s4hana_addon and http://help.sap.com/ibp_erp_addon.

6. Define datastores in SAP Cloud Integration for data services for the source and target systems. Make sure that you set the connection type to `WebSocket RFC`.
   For more information about configuring datastores with connection type `WebSocket RFC`, see the *Help Center for SAP Cloud Integration for data services* at https://help.sap.com/viewer/product/ SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES under ▌▶ *Datastores* ❯ *Datastore Types and Their Properties* ❯ *SAP Integrated Business Planning via WebSocket RFC* ❐ .

7. Define the mapping content on the SAP Cloud Integration for data services server in the cloud.
   In this step, you map the source data to the flexible planning model in SAP IBP and also map the SAP IBP planning result data back to the target systems. For more information, see Setting Up the Mapping Content [page 11].

2.1.1.1   Defining the Communication Arrangement

Define the communication arrangement based on the `SAP_COM_0593` communication scenario. The authentication method can be either certificate-based using an x.509 certificate or basic authentication.

Using Certificate-Based Authentication (Recommended)

If you want to set up certificate-based authentication, make sure that the client certificate you use is signed by a certification authority accepted for inbound integration. For more information, see SAP Note 2871840 ☁ .

To define the communication arrangement with a certificate-based authentication (recommended), proceed as follows:

1. Create a personal security environment (PSE) file and upload the private key of the client certificate to SAP Data Services Agent. For more information, see the *SAP Data Services Agent Guide* guide at https://help.sap.com/viewer/product/SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES under ▌▶ *Setting Up a WebSocket RFC Connection* ❯ *Connecting to SAP IBP Using Certificate-Based Authentication* ❐ .

2. Upload the public certificate that belongs to the private key to SAP Integrated Business Planning for Supply Chain as follows:
   1. Log in to your SAP IBP system.
   2. Open the *Maintain Communication Users* app, and choose *New*.
   3. Enter a username and description, and choose *Save*.
   4. In the *Certificate* section, choose *Upload Certificate*.
      Select the .CRT file that you have generated.
   5. Choose *Save*.

3. Navigate to the *Communication Systems* app, and choose *New*.

4. Enter the system ID and name for your communication system.

5. Under *General* in the *Technical Data* section, select *Inbound Only*.

6. Under *Users for Inbound Communication*, add the communication user you have previously created and select the authentication method.

7. Choose *Save*.

8. Navigate to the *Communication Arrangements* app, and choose *New*.
   Select the communication scenario *Planning - Data Service via WebSocket RFC Integration SAP_COM_0593*, and enter a name for your communication arrangement.
9. Select the communication system that you have previously created, and choose *Save*.

Using Basic Authentication

To define the communication arrangement with basic authentication, do as follows:

1. Log in to your SAP IBP system.
2. Open the *Maintain Communication Users* app, and choose *New*.
3. Enter a username, description, and password, and choose *Save*.
4. Navigate to the *Communication Systems* app, and choose *New*.
5. Enter the system ID and name for your communication system.
6. Under *General* in the *Technical Data* section, select *Inbound Only*.
7. Under *Users for Inbound Communication*, add the communication user you have previously created and select the authentication method.
8. Choose *Save*.
9. Navigate to the *Communication Arrangements* app, and choose *New*.
   Select the communication scenario SAP_COM_0593, *Planning - Data Service via WebSocket RFC Integration*, and enter a name for your communication arrangement.
10. Select the communication system that you have previously created, and choose *Save*.

3 Setting Up the Mapping Content

For every system, file share, and database you want to communicate with, maintain a datastore in SAP Cloud Integration for data services. The datastore contains the technical communication parameters as well as the source and target structures for the mapping.

Afterwards, you can set up one or more integration projects. The following diagram shows the workflow for setting up your mapping content:
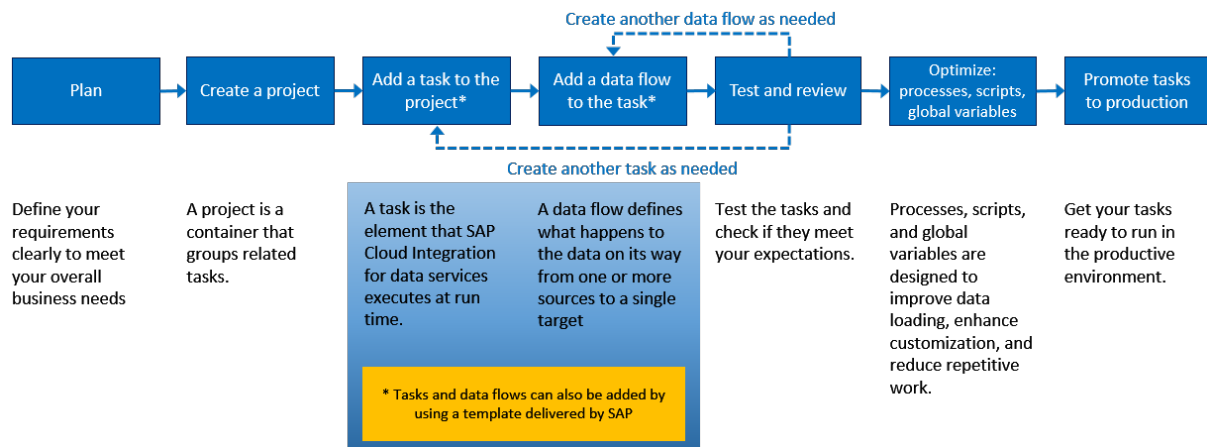
The templates described in the following chapters help you set up a workflow that fulfills your business requirements. You can copy the templates to your own tasks and then adapt them to your needs. The details on how to adapt the template are described in the step-by-step instructions.

Related Information

4 Switching from JDBC to WebSocket RFC Connection

The switch from JDBC to WebSocket RFC is a technical change that affects the communication between SAP Integrated Business Planning for Supply Chain (SAP IBP) and SAP Cloud Integration for data services.

> **i Note**
>
> This change is only relevant for SAP IBP systems that were set up in version 2105 or earlier.
>
> If you're unsure whether WebSocket RFC connection has been set up for your system, go to the *Display Communication Scenarios* app and search for the scenario `ID SAP_COM_0593` or the scenario name *Planning - Data Service via WebSocket RFC Integration*. If you can't find either, your system is still using JDBC and needs to be adapted.

This switch involves making some changes to your data integration setup in the SAP Data Services Agent and SAP Cloud Integration for data services server, and might also require adjustments on data flow level.

The following interactive image shows an overview of the process of setting up WebSocket RFC connection with certificate-based authentication. Hover over each area for a description. Click highlighted areas for more information.

Switching to Websocket RFC Connection

| Preparation | Steps in SAP IBP | Steps in DS Agent | Steps in CI-DS | Finalization |
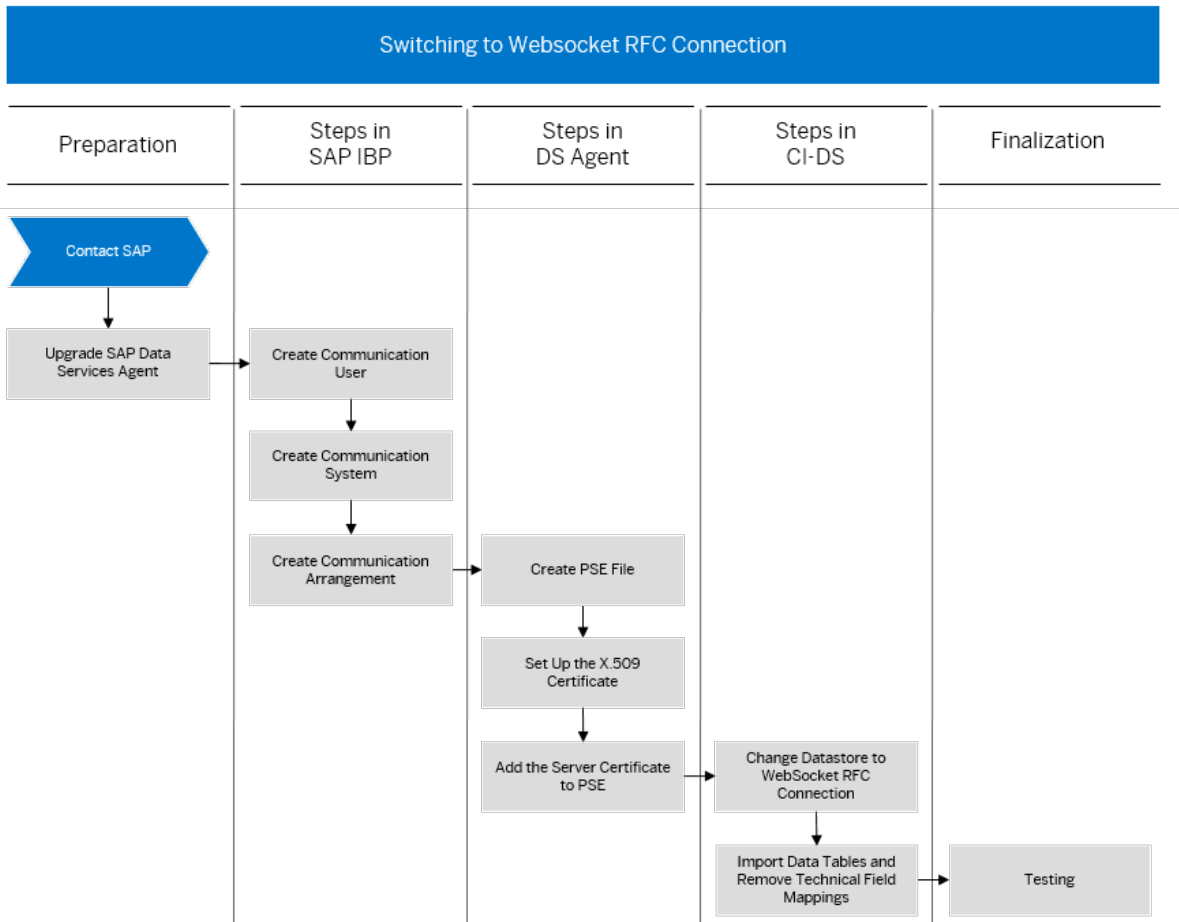
- Defining the Communication Arrangement [page 16]
- Switching from JDBC to WebSocket RFC Connection [page 12]
- Switching from JDBC to WebSocket RFC Connection [page 12]
- Switching from JDBC to WebSocket RFC Connection [page 12]
- Switching from JDBC to WebSocket RFC Connection [page 12]
- Cleaning Up Table Columns [page 18]

> i Note
>
> The migration takes place in consecutive waves. Each wave consists of four weeks; we recommend to reserve the first two weeks for testing in the test system and second two weeks for testing in the productive system. When you're ready, apply for the migration by filling and submitting the following survey: Survey to switch over from JDBC to WebSocket RFC ➦ .

Benefits

This communication protocol provides the same set of features as JDBC, and the following additional benefits:

- SAP Data Services Agent communicates directly with the SAP IBP database instead of going via the SAP Cloud Integration for data services server. This simpler architecture increases system stability. and makes it easier to find the root cause if an issue occurs.
- Data extraction is not limited to 200/400m records in one flow, so it's no longer necessary to split tasks or create multiple tasks due to system limitations.
- WebSocket RFC supports X.509 certificate-based authentication, which is recommended for productive systems and requires a personal security environment (PSE).
- Stability and reliability for long-running jobs are improved. As there is direct connection between SAP IBP and SAP Data Services Agent, there are no more time-out errors for jobs when they need to query initial data for more than 15 minutes.
- Systems using a WebSocket RFC connection are suitable for hyperscaler deployment. This leads to more elasticity and speed.
- As opposed to the system setup using the SAP-managed communication arrangement, the new configuration supports the IN and NOT IN operators in filter conditions used when exporting data from SAP IBP using SAP Cloud Integration for data services. For more information about differences in filtering, see *Data Export from SAP Integrated Business Planning for Supply Chain* at http://help.sap.com/ibp under ▶ *Configuring Data Export in* ❭ SAP Cloud Integration for data services ▶ *Best Practices for Extracting Data from SAP IBP* ❭.

Prerequisites

You are using SAP Data Services Agent version 1.0.11 patch level 42 or higher. We recommend upgrading to the latest agent version at least two weeks before the switch.

> ⚠ Caution
>
> Do not develop new tasks in SAP Cloud Integration for data services during the migration process. The test system will already be running on WebSocket RFC, while the production system will still be running on JDBC, and tasks, including the meta data associated with them, should not be promoted from WebSocket RFC to JDBC.

> ⚠ Caution
>
> There should be no system refresh from the production to the quality/test system during the migration process, as this would disrupt the migration configuration in the test system.

Known Restrictions

Having a system setup with WebSocket RFC connection has the following known restrictions:

- The use of certain technical fields is not supported with WebSocket RFC connection, therefore, you need to clean up your data tables and mappings. For more information, see Cleaning Up Table Columns [page 18].
- On the read interface, decreasing the length of a field is the only technical change that is accepted in SAP IBP without re-importing the metadata in SAP Cloud Integration for data services. After changing the type, the number of decimal places, or increasing the length of a field in SAP IBP, you have to import the relevant table again in SAP Cloud Integration for data services under ▶ *Datastores* ❯ *Tables* ▶. You might also need to adapt the data flow to ensure that the field with the correct length and type is used.
- If you map more than one PERIODIDn fields, or there are more than one PERIODIDn fields in your sort and filter criteria, note that time aggregation will be based on the lowest requested time period level.
- If neither timestamp nor PERIODIDn fields are mapped, SAP IBP returns the data aggregated to the whole planning horizon. In case the timestamp field is mapped, but PERIODIDn is not, an error occurs.

Related Information

For the end-to-end process of setting up the connection with certificate-based authentication and the latest updates, see 3152253 . For setting up basic authentication, see 3166596 .

4.1 Changing the Data Integration Setup

In order to change the connection between your systems, you need to modify the integration setup.

Prerequisites

You are using SAP Data Services Agent version 1.0.11 patch level 42 or higher. For information about installing the latest version, see the *SAP Data Services Agent Guide* at https://help.sap.com/viewer/product/SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES.

Procedure

To enable connection between your systems via WebSocket RFC, change the setup of data integration with SAP Cloud Integration for data services as follows:

1. In SAP IBP, activate your planning areas. Check the application log to make sure that application tables and staging tables have been generated successfully.

2. Define the communication arrangement in SAP IBP based on the `SAP_COM_0593` communication scenario. The authentication method can be either certificate-based or basic authentication.
   For more details on setting up the communication arrangement, see Defining the Communication Arrangement [page 16].
3. In SAP Cloud Integration for data services, set the connection type of datastores representing your source and target systems to `WebSocket RFC`.
   For more information about configuring datastores with connection type `WebSocket RFC`, see the *Help Center for SAP Cloud Integration for data services* at https://help.sap.com/viewer/product/ SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES under ▐▶ *Datastores* ❯ *Datastore Types and Their Properties* ❯ *SAP Integrated Business Planning via WebSocket RFC* ▐ .
4. In SAP Cloud Integration for data services, import the metadata of your tables and views again, and run your integration jobs. In case there are errors caused by missing technical columns, remove the mapping of those fields from your data flows. For more information, see Cleaning Up Table Columns [page 18].
5. Test your data integration jobs thoroughly.

4.1.1 Defining the Communication Arrangement

Define the communication arrangement based on the `SAP_COM_0593` communication scenario. The authentication method can be either certificate-based using an x.509 certificate or basic authentication.

Using Certificate-Based Authentication (Recommended)

If you want to set up certificate-based authentication, make sure that the client certificate you use is signed by a certification authority accepted for inbound integration. For more information, see SAP Note 2871840 .

> **i Note**
>
> For the end-to-end process of setting up the connection with certificate-based authentication and the latest updates, see 3152253 .

To define the communication arrangement with a certificate-based authentication (recommended), proceed as follows:

1. Create a personal security environment (PSE) file and upload the private key of the client certificate to SAP Data Services Agent. For more information, see the *SAP Data Services Agent Guide* guide at https://help.sap.com/viewer/product/SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES under ▐▶ *Setting Up a WebSocket RFC Connection* ❯ *Connecting to SAP IBP Using Certificate-Based Authentication* ▐ .
2. Upload the public certificate that belongs to the private key to SAP Integrated Business Planning for Supply Chain as follows:
   1. Log in to your SAP IBP system.
   2. Open the *Maintain Communication Users* app, and choose *New*.
   3. Enter a username and description, and choose *Save*.
   4. In the *Certificate* section, choose *Upload Certificate*.

Select the .CRT file that you have generated.

    5. Choose *Save*.

3. Navigate to the *Communication Systems* app, and choose *New*.

4. Enter the system ID and name for your communication system.

5. Under *General* in the *Technical Data* section, select *Inbound Only*.

6. Under *Users for Inbound Communication*, add the communication user you have previously created and select the authentication method.

7. Choose *Save*.

8. Navigate to the *Communication Arrangements* app, and choose *New*.
Select the communication scenario *Planning - Data Service via WebSocket RFC Integration SAP_COM_0593*, and enter a name for your communication arrangement.

9. Select the communication system that you have previously created, and choose *Save*.

Using Basic Authentication

> i Note
>
> For the end-to-end process of setting up the connection with basic authentication, see 3166596 .

To define the communication arrangement with basic authentication, do as follows:

1. Create a personal security environment (PSE) file. For more information, see the *SAP Data Services Agent Guide* guide at https://help.sap.com/viewer/product/ SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES under ▶ *Setting Up a WebSocket RFC Connection* ▶ *Connecting to SAP IBP Using Password Authentication* ▶.

2. Log in to your SAP IBP system.

3. Open the *Maintain Communication Users* app, and choose *New*.

4. Enter a username, description, and password, and choose *Save*.

5. Navigate to the *Communication Systems* app, and choose *New*.

6. Enter the system ID and name for your communication system.

7. Under *General* in the *Technical Data* section, select *Inbound Only*.

8. Under *Users for Inbound Communication*, add the communication user you have previously created and select the authentication method.

9. Choose *Save*.

10. Navigate to the *Communication Arrangements* app, and choose *New*.
Select the communication scenario SAP_COM_0593, *Planning - Data Service via WebSocket RFC Integration*, and enter a name for your communication arrangement.

11. Select the communication system that you have previously created, and choose *Save*.

4.1.2 Cleaning Up Table Columns

After changing the connection type of your datastores representing your SAP IBP source and SAP IBP target systems to **WebSocket RFC**, technical columns must be removed from data tables and data flow mappings.

Tables and views in datastores that have connection type **JDBC** include technical columns that are not used in data integration. After switching to WebSocket RFC connection and updating the metadata of your tables and views, columns get cleaned up, however, having them mapped in data flows can cause errors.

Errors are most likely to occur in the following table types:

- Data export scenario:
  - Master data core tables
  - Calculation scenario tables
    In the case of calculation scenario tables, errors are usually caused by the mapping of `PERIODID` columns. If your data flows contain `PERIODID` in their data mapping, remove it, and instead, map the `PERIODIDn` column that corresponds to the appropriate time aggregation level.
- Data import scenario:
  - Master data staging tables
  - Key figure staging tables

First, import the metadata of tables in SAP Cloud Integration for data services as follows:

1. Navigate to the *Datastores* view.
2. Select the datastore you want to clean up.
3. On the *Tables* tab, choose *Import Objects*.
4. Select the tables that are already marked with a checkmark in the *Imported* column, and choose *Import*.
5. Repeat these steps for all datastores of which you've changed the connection type.

Now that technical columns are removed from the updated metadata of data tables, **trigger data integration** for all your data flows. In case you have technical columns left in the mapping of a data flow, the job will fail with an error such as the following:

> **⁂ Example**
>
> ```
> <Date> <time> (E)(19306:1223608128)RES-20108 |SESSION <SESSION_ID>|DATAFLOW
> <DATA_FLOW_ID>|STATEMENT <GUID::<GUID> QUERY Target_Query> Column <CURRNUM> was
> not found in table <SOPMD_CURRENCY>. Check the spelling of the referenced table
> and column and ensure that the table exists and such a column exists in the
> table.
> ```

To resolve the error, remove the mapping of the column that is specified in the error message as follows:

1. In SAP Cloud Integration for data services, navigate to the *Projects* view.
2. Open the task, then open the data flow for editing where the error occurred.
3. In the data flow editor, open the source object.
4. In the list of columns, select the ones that are listed in the error message.
5. In the *Transform Details* section, open the *Mapping* tab.
6. Remove all text from the input field.

> **i Note**
>
> If the target of the data flow is a file, all columns of your file must be mapped. If a technical column of the output file would become unmapped, remove the column from the file under *Datastores*. In case you still want to keep the column, provide a constant value as mapping data in the data flow.

5 Enabling Improved Master Data Processing

The way master data integration and processing takes place has changed, and if your system is already configured correctly, the new features were enabled automatically in version 2211. However, if your system is not ready for the change yet, data integration stays as it is for you until you contact SAP and ask for the switch.

> ℹ Note
>
> If your SAP IBP system was set up in version 2205 or higher, this feature is enabled for you by default and there are no further changes required.

Improved data processing simplifies the use of flexible master data, which is available in order-based planning (OBP) using real-time integration. It allows you to keep your data in sync without having to run a synchronization job after integrating flexible master data to SAP IBP using SAP Cloud Integration for data services. For more information about the *Order-Based Planning: Master Data Synchronization* application job, see the application help at http://help.sap.com/ibp under ▶ *Business Applications* ▶ *Order-Based Planning* ▶ *Using Order-Based Planning with Flexible Master Data* ▶ *Setting Up and Refining Plans* ▶ *Basic Settings* ▶ *Integrating Master Data and Transactional Data* ▶ *Synchronization of Master Data Configured for Order-Based Planning* ◀.

As using this feature with certain incorrect configurations could lead to errors, you only get it automatically if the configuraton of your system allows it. If you want to have it enabled, you first need to make sure that the following prerequisites are met:

- The configuration of attributes as key figures must be in alignment with the guidelines. Make sure that the non-time root attributes of the key figure's base planning level and the key attributes of the master data type for this configuration are exactly the same.
  For example, if the master data type for the attribute as key figure has the key attributes PRDID, LOCID, and CUSTID, the base planning level of the key figure needs to have the non-time root attributes PRDID, LOCID, and CUSTID. Selecting a base planning level that only has PRDID and LOCID as non-time root attributes would lead to issues during data integration and inconsistent data.
  Instead of using attributes of key figures, you can also load key figure data directly.
  For more information, see the *Model Configuration Guide* at http://help.sap.com/ibp under *Attributes as Key Figures*.
- Before uploading master data to a version where version-specific attributes as key figures are configured, make sure that the target version uses version-specific master data types as well. If it is not the case, attributes as key figures will only be updated in the base version. To check your settings, open your planning area in the *Planning Areas* app, navigate to the relevant version, and make sure that the *Use Version-Specific Master Data Types* checkbox is selected.
- In case you want to integrate master data both into OBP (external master data) and time-series-based planning areas, create separate tasks for them. Otherwise, the job will fail.
- Make sure that scheduled jobs for uploading time periods in SAP Cloud Integration for data services use the batch command *REPLACE*. Otherwise, the jobs will fail.

> **i Note**
>
> Message numbers related to time-series-based data integration will be changed both in reporting tables that you can export from SAP Cloud Integration for data services and rejection reports that you can download from the *Data Integration Jobs* app. In case you have processes that rely on message numbers, these processes will be affected.

After finalizing your configuration, create an incident under component `SCM-IBP-INT-TS` to have the improved data processing enabled for your system. Please note that apart from the configuration of attributes as key figures, SAP cannot check if the prerequisites are met, and after the switch, the processes listed above will fail if not adapted as described.

To check whether you've already received it, integrate master data into SAP IBP and check the log messages. If your system is already using the new codeline, there is the following entry in the logs: *Improved master data processing is enabled*.

> **i Note**
>
> Note that enabling the improved data processing might be a prerequisites of future features of data integration.

Related Information

See the application help at http://help.sap.com/ibp under ▌▶ *Business Applications* ❯ *Order-Based Planning* ❯ *Using Order-Based Planning with Flexible Master Data* ❯ *Setting Up and Refining Plans* ❯ *Basic Settings* ❯ *Integrating Master Data and Transactional Data* ❯ *Integrating Data in the OBP Master Data Store* ▌.

6 About the Templates

Templates in SAP Cloud Integration for data servicesSAP Cloud Integration for data services provide predefined content which serves as the starting point for populating your SAP Integrated Business Planning applications, or for transferring data from SAP Integrated Business Planning to another SAP application, such as SAP Advanced Planning and Optimization (SAP APO).

The templates are designed to meet the specific requirements of SAP Integrated Business Planning data, and reduce the time needed to get up and running with the application.

Three types of templates are available in SAP Cloud Integration for data services:

- General purpose templates
- Master data templates
- Key figure templates

General purpose templates contain information required by SAP Integrated Business Planning to process the data after it is loaded. The general purpose templates, `SOP_APO_Task` and `SOP_ECC_Task`, contain global variables, preload scripts and postload scripts. After using either of these templates to create a task, you add datastore information and your own data flows.

Tasks created from the master data templates and key figure templates contain the data flows, global variables, and scripts necessary to do the following:

- Tasks that transfer data from an SAP application to SAP Integrated Business Planning:
  - Extract data from SAP applications sources (SAP ERP and SAP APO)
  - Transform it as required for SAP Integrated Business Planning
  - Load the data to predefined target tables for the default model in the staging area of SAP Integrated Business Planning
- Tasks that transfer data from SAP Integrated Business Planning to a SAP APO:
  - Extract data from SAP Integrated Business Planning
  - Transform it as required for SAP APO
  - Call a Web service that will load the data to the SAP APO/SAP SCM system
  - Write a log

> **i Note**
>
> SAP Advanced Planning and Optimization can be deployed as part of an SAP Supply Chain Management (SAP SCM) installation, and as an add-on to SAP ERP. Therefore, when systems are mentioned, SAP APO/SAP SCM system refers to the system in which the SAP APO application is running.

> **i Note**
>
> The following templates in SAP Cloud Integration for data services access the database of the source system directly via generated ABAP coding:
>
> - IBP_KF_DemandPlanning
> - IBP_MD_PlanningArea

- SOP_MD_LocationMaster
- SOP_MD_ProductMaster
- SOP_MD_LocationProd
- SOP_MD_CustomerMaster
- SOP_MD_Resource
- SOP_KF_Actuals
- SOP_KF_Inventory
- SOP_KF_CapacityLimit
- IBP_KF_Actuals
- IBP_KF_InitialInventory
- IBP_KF_SafetyStock
- IBP_KF_OpenOrders_S4
- IBP_KF_OpenOrders_ERP
- IBP_KF_PlannedPrice

Please ensure that this type of access is covered by your database license.

Task Elements

Each task contains the elements described in the following table:

Element	Description
Source datastore	Connects SAP Cloud Integration for data services and the source system, such as SAP ERP
Target datastore	Connects SAP Cloud Integration for data services and the target system, such as SAP Integrated Business Planning
Target table	Target table of the data load process
Data flow	Defines the movement and transformation of data from source to target. A data flow may combine data from several source tables or extractors, but it has exactly one target table.
Global variables	Global variables allow you to parameterize a task for execution without changing its definition. Typical examples of global variables are selection parameters.
Preload script	Initializes global variables required for task run
Postload script	Records the timestamp of the most recent run

Advantages of Using Templates

Each SAP Integrated Business Planning implementation is unique, therefore you are likely to customize the out-of-the-box tasks and data flows that you create from the SAP Integrated Business Planning templates.

However, using the templates to create tasks and data flows as a starting point has the following advantages over creating tasks from scratch:

- Template tasks provide the framework and thus reduce the integration development time.
- Template tasks contain the global variables required to successfully load and process data in SAP Integrated Business Planning.
- Preload scripts are configured with default values for these global variables and provide a framework. This eliminates the need to write preload scripts from scratch.
- Data flows in master data templates and key figure templates follow best practice design for optimal integration performance.
- Data flows in master data templates and in key figure templates contain mappings from SAP ERP and SAP SCM/SAP APO source systems or to SAP SCM/SAP APO target systems, and include joins and predefined filters.

> i Note
>
> To learn about known issues when using SAP Cloud Integration for data services with SAP Integrated Business Planning, see SAP Notes 1826078 and 2007254 .

7 Templates for SAP Integrated Business Planning for Supply Chain

> **i Note**
>
> SAP Cloud Integration for data services may have more template tasks than those listed in this guide. However, only the tasks listed in this guide are relevant for SAP Integrated Business Planning for Supply Chain (SAP IBP).

> **i Note**
>
> The delivered templates are assigned to a specific sample planning area and its staging tables. They can be compatible with additional sample planning areas. If they are not compatible with the planning area, they will not work or will need to be adapted. For a list of available sample planning areas, search for "Sample Planning Areas" in the latest version of SAP IBP product documentation on the Help Portal at https://help.sap.com/ibp.

> **i Note**
>
> If we mention in this document that a template is compatible with SAP S/4HANA, we always mean SAP S/4HANA on premise. For S/4HANA cloud, the used integration technology is not supported.
>
> If we mention below that a template is compatible with SAP ERP, we mean SAP ECC 6.0, Enhancement Package 4 and higher.

> **i Note**
>
> SAP IBP accepts key figure data having a time stamp with seconds precision. Although you can generate more precise time stamps with SAP Cloud Integration for data services, they will be truncated.

You can use the following tasks as templates when you develop your tasks to fit the specific needs of your company:

Master Data Templates to Transfer Data to SAP Integrated Business Planning for Supply Chain

These templates serve to transfer master data from the source system (SAP S/4HANA on premise, SAP ERP, or SAP APO) to SAP Integrated Business Planning for Supply Chain applications.

You can load master data into SAP Integrated Business Planning for Supply Chain both from SAP S/4HANA on premise, SAP ERP, and SAP APO. Make sure you import master data from the system that is the leading system for master data in your landscape.

> **i Note**
>
> While sales orders and outbound deliveries are considered transactional data in SAP S/4HANA on premise, SAP ERP, in SAP Integrated Business Planning for Supply Chain for demand, they are modeled as master data types.

Task Name	Purpose	SAP Applications Source	Relevant for SAP IBP Application	Target Sample Planning Area
IBP_MD_S4_ERP_AddOn	To transfer product, location, resource, location product, and resource location from S/4 or ERP Add-On to IBP	SAP S/4HANA SAP ERP (with Add-On for IBP)	All, except order-based operational planning	All except SAP7, SAP8
IBP_DDR_MD_ERP_AddOn	To extract master data from SAP ECC and to load it to the demand-driven replenishment planning area of SAP IBP	SAP S/4HANA SAP ERP (with Add-On for IBP)	Demand-Driven Replenishment	SAP8
SOP_MD_LocationMaster	To transfer location master data	SAP S/4HANA SAP ERP	Sales and Operations, Demand	All except SAP7, SAP8
SOP_MD_ProductMaster	To transfer product master data	SAP S/4HANA SAP ERP	Sales and Operations, Demand	All except SAP7, SAP8
SOP_MD_LocationProd	To transfer location product master data	SAP S/4HANA SAP ERP	Sales and Operations	All except SAP7, SAP8
SOP_MD_CustomerMaster	To transfer customer master data	SAP S/4HANA SAP ERP	Sales and Operations, Demand	All except SAP7, SAP8
SOP_MD_Resource	To transfer resource master data	SAP S/4HANA SAP ERP	Sales and Operations	All except SAP7, SAP8
IBP_MD_PlanningArea	To extract product master data, unit of measure, unit of measure conversion factors, location master data, and customer master data from SAP APO, and load it to SAP IBP	SAP APO	All, except order-based operational planning	All except SAP7, SAP8

Task Name	Purpose	SAP Applications Source	Relevant for SAP IBP Application	Target Sample Planning Area
IBP_MD_SalesOrder _InitialLoad	To transfer sales order data, initial load	SAP S/4HANA SAP ERP	Demand	All except SAP7, SAP8
IBP_MD_SalesOrder _DeltaLoad	To transfer sales order data, delta load	SAP S/4HANA SAP ERP	Demand	All except SAP7, SAP8
IBP_MD_Deliveries _InitialLoad	To transfer outbound delivery data, initial load	SAP S/4HANA SAP ERP	Demand, Unified Planning Area	All except SAP7, SAP8
IBP_MD_Deliveries _DeltaLoad	To transfer outbound delivery data, delta load	SAP S/4HANA SAP ERP	Demand, Unified Planning Area	All except SAP7, SAP8

Master Data Templates to Transfer Data Within SAP Integrated Business Planning for Supply Chain

These templates serve to transfer master data between different planning areas within SAP Integrated Business Planning for Supply Chain.

> **i Note**
>
> While both transactional and master data are collected by the templates, these transactional data are stored as master data attributes in SAP7.

Task Name	Purpose	Source Sample Planning Area	Relevant for SAP IBP Application	Target Sample Planning Area
IBP_DDR_OBP_KF_to _File	To transfer relevant demand-driven replenishment key figures and master data attributes to a file to integrate them into the OBP planning area as master data attributes	SAP7	Order-Based Planning, Demand-Driven Replenishment	SAP8
IBP_DDR_OBP_MD_fr om_File	To extract data from CSV file and transfer it to OBP planning area	SAP7	Order-Based Planning, Demand-Driven Replenishment	SAP8

Key Figure Templates to Transfer Data to the IBP Unified Planning Area

These templates serve to transfer key figure values from the source system (SAP S/4HANA on premise, SAP ERP or SAP APO) to SAP Integrated Business Planning for Supply Chain. The target is the Unified Planning Area SAPIBP1 for all templates. These templates implement additional mapping content, global variables, and preload script content.

Task Name	Purpose	SAP Applications Source
IBP_KF_S4_ERP_AddOn	To extract key figures from SAP S/4HANA on-premise or SAP ECC and to load it to SAP IBP	SAP S/4HANA on premise, SAP ERP
IBP_KF_Actuals	To transfer actuals quantity and revenue key figure data from SAP S/4HANA on premise, SAP ERP to the unified planning area	SAP S/4HANA on premise, SAP ERP
IBP_KF_InitialInventory	To transfer inventory key figure data from SAP S/4HANA on premise, SAP ERP to the unified planning area	SAP S/4HANA on premise, SAP ERP
IBP_KF_OpenOrders_ERP	To transfer open orders quantity and revenue key figure data from SAP ERP to the unified planning area	SAP ERP
IBP_KF_OpenOrders_S4	To transfer open order key figure data from SAP S/4HANA on premise to the unified planning area	SAP S/4HANA on premise
IBP_KF_PlannedPrice	To transfer planned price key figure data from SAP S/4HANA on premise, SAP ERP to the unified planning area	SAP S/4HANA on premise, SAP ERP
IBP_KF_SafetyStock	To transfer safety stock key figure data from SAP S/4HANA on premise, SAP ERP to the unified planning area	SAP S/4HANA on premise, SAP ERP
IBP_to_ERP_PIR_via_RFC IBP_to_ERP_PIR_via_RFC_2108	To transfer consensus demand quantity key figure data from the unified planning area to SAP S/4HANA on premise, SAP ERP	SAP S/4HANA on premise, SAP ERP

Key Figure Templates to Transfer Data to Application-Specific Planning Areas in SAP IBP

These templates serve to transfer key figure values from the source system (SAP S/4HANA on premise, SAP ERP, or SAP APO) to SAP IBP applications. Please note that new templates have been added that are compatible with the Unified Planning Area.

Task Name	Purpose	SAP Applications Source	Relevant for SAP IBP Application
IBP_DDR_KF_ERP_AddOn	To extract key figures from SAP S/4HANA on-premise or SAP ECC and to load it to a demand-driven replenishment planning area of SAP IBP copied from the SAP8 example planning area	SAP S/4HANA on premise, SAP ERP	SAP8 (Demand-Driven Replenishment)
SOP_KF_Actuals	To transfer actuals key figure data from SAP S/4HANA on premise, SAP ERP	SAP S/4HANA on premise, SAP ERP	SAP2 (Sales and Operations and Supply)
SOP_KF_Inventory	To transfer inventory key figure data from SAP S/4HANA on premise, SAP ERP	SAP S/4HANA on premise, SAP ERP	SAP2 (Sales and Operations and Supply)
SOP_KF_OpenOrders	To transfer open orders key figure data from SAP S/4HANA on premise, SAP ERP	SAP S/4HANA on premise, SAP ERP	SAP2 (Sales and Operations and Supply)
SOP_KF_SalesForecastPrice	To transfer sales forecast price key figure data from SAP S/4HANA on premise, SAP ERP	SAP S/4HANA on premise, SAP ERP	SAP2 (Sales and Operations and Supply)
SOP_KF_CapacityLimit	To transfer capacity limit key figure data from SAP APO	SAP APO	SAP2 (Sales and Operations and Supply)
SOP_KF_Consumption	To transfer capacity consumption key figure data from SAP APO	SAP APO	SAP2 (Sales and Operations and Supply)
IBP_KF_DemandPlanning	To transfer demand planning key figures, such as consensus demand plan	SAP APO	SAP2, SAP6, SAPIBP1

Key Figure Templates to Transfer Data to Other SAP Applications

The following template serves to transfer sensed demand data (short-term forecasting data created by demand sensing) from SAP Integrated Business Planning for Supply Chain for demand to Supply Network

Planning in SAP APO, or to transfer data from a demand-driven replenishment planning area in SAP Integrated Business Planning for Supply Chain to another system (such as SAP ERP).

Task Name	Purpose	SAP Target System	Source Planning Area
IBP_DDR_to_ERP_AddOn_via_RFC	To transfer the input for the MRP run in ERP from the demand-driven replenishment planning area in SAP IBP to the IBP add-on in SAP ECC	SAP S/4HANA on premise, SAP ERP	SAP8 (Demand-Driven Replenishment)
IBP_DDR_to_S4_Buffer_Integration_via_API	To transfer the daily buffers using top of red, top of yellow and top of green to an SAP S/4HANA system	SAP S/4HANA	SAP8 (Demand-Driven Replenishment)
IBP_DDR_to_S4_DCP_via_API	To transfer the MRP type and initial buffers to the material master for decoupling points in SAP S/4HANA, or revert the MRP type for location materials that are no longer decoupling points	SAP S/4HANA	SAP8 (Demand-Driven Replenishment)
IBP_KF_PlannedIndependentRequirements	To transfer demand sensing data as planned independent requirements to Supply Network Planning	SAP APO	SAP6 (Demand)
IBP_KF_POSSales	To transfer POS sales data from SAP Demand Signal Management	SAP DSiM	SAP6 (Demand)
IBP_KF_POSStock	To transfer POS stock data from SAP Demand Signal Management	SAP DSiM	SAP6 (Demand)

7.1   SAP ERP or SAP S/4HANA Add-On Integration

Prerequisites for Creating a Task Based on the SAP IBP Add-On for SAP S/4HANA or SAP ERP

1. Install and configure one of the following add-ons on the source system (including the latest support package):

- SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_s4hana_addon)
- SAP ERP, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_erp_addon)

2. Perform the basic configuration of the add-on, for example maintain the plants relevant for SAP IBP integration. This also includes activating the business transaction event *IBP* in order to achieve automatic updates of most of the replication tables used for extraction whenever the underlying master data has been changed.

3. Run the initial load transactions `/IBP/ECC_INT_LOC`, `/IBP/ECC_INT_MAT`, `/IBP/ECC_INT_WC`, `/IBP/ECC_INT_PDS`, `/IBP/ECC_INT_TLANE`, and `/IBP/ECC_INT_TRANS`. This will fill the replication tables `/IBP/LOC_EXT`, `/IBP/MARA_EXT`, `/IBP/MAKT_EXT`, `/IBP/MARC_EXT`, `/IBP/MARM_EXT` `/IBP/RESHEAD_EXT`, `/IBP/RESDESC_EXT`, `/IBP/SOSHEAD_EXT`, `/IBP/SOSBOM_EXT`, `/IBP/TLANE_EXT`, `/IBP/ORDER_EXT`, and `/IBP/STOCK_EXT`.

   You can use transaction **SE16N** (or **SE16**) in the source system to check if the tables are filled correctly. If business transaction event *IBP* is activated, almost all master data changes are automatically updated in the replication tables. This is not true for plants and work centers. After changing plants and work centers, you have to run transaction `/IBP/ECC_INT_LOC` or `/IBP/ECC_INT_WC` again to reflect the changes in the replication tables. See the linked application help of the add-ons for more details. For information on the extension concept for the extractors and the underlying staging tables, see the SAP S/4HANA, Supply Chain Integration Add-On for SAP Integrated Business Planning administrator's guide at http://help.sap.com/ibp_s4hana_addon under *Administration* > *Administrator's Guide - SAP S/4HANA, Supply Chain Integration Add-On for SAP Integrated Business Planning* > *Configuration for Time-Series-Based Integration* > *Extensions for Time-Series-Based Integration* and the SAP ERP, Supply Chain Integration Add-On for SAP Integrated Business Planning administrator's guide at http://help.sap.com/ibp_erp_addon under *Administration* > *Administrator's Guide - SAP ERP, Supply Chain Integration Add-On for SAP Integrated Business Planning 1.1* > *Configuration for Time-Series-Based Integration* > *Extensions for Time-Series-Based Integration* .

4. Activate the involved extractors in the source system. This is done as follows:

   1. Log on to a client that allows changes to cross-client Customizing. If you want to transport the changes to follow-on systems, you also need a transport request or the authorization to create one.

   2. Run transaction **RSA9** to copy the delivered extractor hierarchy structure to the active one. Please be aware that this step overwrites all modifications that have been made to the active extractor hierarchy. After this step, the extractor hierarchy contains the following entries:
      SCM-IBP-INT (SCM: Integration of SAP Integrated Business Planning)
      - `SCM-IBP-INT-ECC-TS-IO` (IBP Time-Series Based ERP Integration: Master Data)
        - `/IBP/LOCATIONPRODUCT_ATTR` (IBP Location Product Attributes)
        - `/IBP/LOCATION_ATTR` (IBP Location Attributes)
        - `/IBP/PRODUCTIONSOURCEITEM_ATTR` (IBP Production Source Item Attributes)
        - `/IBP/PRODUCT_ATTR` (IBP Product Attributes)
        - `/IBP/PRODUCT_TEXT` (IBP Product Descriptions)
        - `/IBP/PRODUCT_UOM_ATTR` (IBP Product Unit of Measure Attributes))
        - `/IBP/RESOURCELOCATION_ATTR` (IBP Resource Location Attributes)
        - `/IBP/RESOURCE_ATTR` (IBP Resource Attributes)
        - `/IBP/RESOURCE_TEXT` (IBP Resource Text)
        - `/IBP/SOURCECUSTOMER_CI_ATTR` (Customer Source (Customer Implementation))

- `/IBP/SOURCELOCATION_ATTR` (IBP Source Location Attributes)
- `/IBP/SOURCEPRODUCTION_ATTR` (IBP Source Production Attributes)
- `/IBP/UNIT_OF_MEASURE_ATTR` (IBP Unit of Measure Attributes)
- `/IBP/UNIT_OF_MEASURE_TEXT` (IBP Unit of Measure Texts)
- `SCM-IBP-INT-ECC-TS` (IBP Time-Series Based ERP Integration: Transactional Data)
  - `/IBP/ACTUALS_QTY_CI_KF IBP` (Actuals Quantity Key Figure)
  - `/IBP/CAPASUPPLY_KF` (Capacity Supply for Resource Location)
  - `/IBP/FORECAST_QTY_CI_KF` (IBP Forecast Quantity Keyfigure (Customer Implemented))
  - `/IBP/ORDER_KF` (Order Quantities aggregated for IBP Key Figures)
  - `/IBP/STOCK_KF` (Stock Quantities aggregated for IBP Key Figures)
  - `/IBP/TOTAL_DEMAND_QTY_CI_KF` (IBP Total Demand Quantity (Forecast and Several Order Types))

If you skip this step, you still find the `SCM-IBP-INT-ECC-TS-IO` and `SCM-IBP-INT-ECC-TS` nodes and their extractors, but they will be located below the node `NODESNOTCONNECTED`.

3. Call transaction **RSA5** and mark the nodes `SCM-IBP-INT-ECC-TS-IO` and `SCM-IBP-INT-ECC-TS` (either using the path ▌ *SAP* ❯ *SCM-IBP-INT* ▐ or ▌ *SAP* ❯ *NODESNOTCONNECTED* ▐) and then click *Activate DataSource*. The system asks for a development package and a transport request.

4. You can test if the extractors have been activated correctly and provide data via the **`/IBP/ETS_REPL_TEST`** transaction. Enter `/N/IBP/ETS_REPL_TEST` in the OK-code, choose ⌨Enter , and in the *Data Sources/Extractor Name* field, enter the extractor name (for example, `/IBP/LOCATION_ATTR`). By default, *Direct Extraction Without ODQ* is set, which runs the extractor in the local session instead of in the background. It gives you the option to debug the selection logic. In lower NetWeaver releases in which only ODP 1.0 is available, the direct extraction does not check if the extractor is exposed. In this case, please switch to *Serial Extraction Using ODQ* mode. If required, you can specify additional selection conditions using the parameter *Selections*. You can also use the *Projections* field to define which fields are requested. This corresponds to fields that are mapped in the SAP Cloud Integration for data services data flow or are used for evaluating filters that cannot be pushed down to the extractor.

The `/IBP/ETS_REPL_TEST` report has a set of delivered system variants. Half of them contain the requested fields and filter settings of the templates delivered in SAP Cloud Integration for data services whereas the other half of the system variants have the same filter settings, but request more fields. They can be used to analyze the details of the selected data. This is especially useful in the case of the `/IBP/ORDER_KF` extractor because the detailed variant shows all the single object IDs and attributes behind the aggregated time series.

> ℹ **Note**
>
> You can also use the `RODPS_REPL_TEST` report for testing, but if you do, please note the following:
>
> - In older releases the report is only available in a simplified version. In this version the field *ODP Context* is not available and the extractor name needs to be entered in the field *Queue Name*. In addition, you only can specify one selection condition for one field using parameters *Field Name*, *Option*, *From value*, and *To value*.
> - The report can also be used for analyzing other extractors, which makes it more flexible, but also more complex.

> **i Note**
>
> For the two text extractors `/IBP/PRODUCT_TEXT` and `/IBP/RESOURCE_TEXT` you need to add at least one selection condition for field `LANGUAGE` to deliver some results.

> **i Note**
>
> As an alternative test tool, you also can use transaction **RSA3**. The report `RODPS_REPL_TEST` follows almost the same access route as SAP Cloud Integration for data services, whereas transaction **RSA3** skips most parts of the framework.

> **i Note**
>
> After applying a support package to the add-on you need to run the steps 4c and 4d again to make ensure that the new and extended extractors are visible.

7.1.1 Creating Datastores and Importing Data Sources and Tables in SAP Cloud Integration for Data Services

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP S/4HANA on premise or SAP ECC system. Provide the following data for the new datastore:

Name and Description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP S/4HANA on premise or SAP ECC system	

2. Import the data sources or extractors that you have activated in your SAP S/4HANA on premise or SAP ECC system as follows:
   1. Select the datastore that represents your SAP S/4HANA or SAP ECC system.
   2. Choose the *Tables* tab if it is not active yet.
   3. Choose *Import Objects*.
   4. Open ▶ *Extractors* ▶ *By Apllication* ▶ *BW Root* ▶ *SCM-IBP-INT* ◀ (SCM: Integration of SAP Integrated Business Planning).
   5. Open the following subfolders:
      - `SCM-IBP-INT-ECC-TS-IO` (IBP Time Series Based ERP Integration: Master Data)
      - `SCM-IBP-INT-ECC-TS` (IBP Time Series Based ERP Integration: Transactional Data)

> **i Note**
>
> Depending on the version of the source system, the subfolders named above with descriptions in brackets might be displayed as GUIDs, however, you can still identify them by their descriptions.

6. Mark all the extractors you want to use to define dataflows in SAP Cloud Integration for data services.
7. Choose *Import*.
8. Choose *OK* without changing the default settings.

> **i Note**
>
> It is also possible to load a single extractor by its name as follows:

1. Select the datastore that represents your SAP S/4HANA or SAP ECC system.
2. Choose *Tables* if it is not active yet.
3. Choose *Import Object by Name*.
4. Select *Extractors* and enter the name of the data source in the *Name* field.
5. Choose *Import*.
6. Choose *OK* without changing the default settings.

3. Create a datastore that represents your SAP Integrated Business Planning system. Provide the following data for the new datastore:

Name and Description

Type	Integrated Business Planning
Instance	Your SAP Integrated Business Planning system

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

7.1.2 Lifecyle Management for Tasks Based on the IBP Add-On for SAP S/4HANA or SAP ERP

The templates `IBP_MD_S4_ERP_AddOn`, `IBP_MD_S4_ERP_AddOn`, `IBP_DDR_MD_ERP_AddOn`, `IBP_DDR_KF_ERP_AddOn`, and `IBP_DDR_KF_to_ERP_AddOn_via_WS` are all based either on the IBP Add-On for S/4HANA or the IBP Add-On for ERP on the one side and the IBP system on the other side. They are updated every release to reflect the latest changes in the add-on and the IBP system.

In general, tasks created using these templates will still work after the IBP system is upgraded or the support package level of the add-on is increased. Regarding incompatible changes, you will be informed upfront by the IBP what's new information.

The upgrade of the IBP system is closely synchronized with the upgrade of the templates, but the upgrade of the add-on is controlled by the customer. Therefore, if you create a new task after an upgrade of the templates

without upgrading the add-on, your task might contain mappings for extractors or fields that do not exist yet in the older version of the add-on.

In that case, you can apply the latest support package of the add-on and reload the extractors in the SAP S/4HANA or SAP ERP datastore. If this is not feasible, you can still use the template after deleting dataflows that use non-existing extractors and deleting field mappings and filter definitions for non-existing fields. You can find information on what has been added in each support package as a comment in the flowgraph of the dataflow.

7.2 Master Data Templates

SAP delivers templates you can use as the starting point for creating your own tasks to transfer data between your SAP ERP, SAP APO, or SAP SCM system and SAP Integrated Business Planning.

> i Note
>
> You can load master data into SAP Integrated Planning for Demand both from SAP ERP and from SAP APO. Make sure you import master data from the system that is the leading system for master data in your landscape.

You must load master data prior to loading key figures.

If you load master data from SAP ERP or SAP S/4HANA on premise we recommend using task `IBP_MD_S4_ERP_AddOn`, which requires installing the latest support package of the IBP add-on in the SAP ERP or SAP S/4HANA system. For more information, see IBP_MD_S4_ERP_AddOn [page 36]. If you choose to use the add-on, please run the tasks in the following order:

1. `IBP_MD_S4_ERP_AddOn`
2. `IBP_MD_SalesOrder_InitialLoad`
3. `IBP_MD_SalesOrder_DeltaLoad`
4. `IBP_MD_Deliveries_InitialLoad`
5. `IBP_MD_Deliveries_DeltaLoad`

If you load master data from SAP ERP without the add-on, perform the tasks in the following order:

1. `SOP_MD_LocationMaster`
2. `SOP_MD_ProductMaster`
3. `SOP_MD_LocationProd`
4. `SOP_MD_CustomerMaster`
5. `SOP_MD_Resource`
6. `IBP_MD_SalesOrder_InitialLoad`
7. `IBP_MD_SalesOrder_DeltaLoad`
8. `IBP_MD_Deliveries_InitialLoad`
9. `IBP_MD_Deliveries_DeltaLoad`

> i Note
>
> In case you want to integrate master data both into order-based planning (OBP) (external master data) and time-series-based planning areas, create separate tasks for them.

> **i Note**
>
> Before you load sales orders or outbound deliveries to SAP Integrated Business Planning, make sure you have transferred all the necessary master data (products, locations, and customers).

7.2.1 IBP_MD_S4_ERP_AddOn

The `IBP_MD_S4_ERP_AddOn` template contains data flows to extract product, location, resource, location product, and resource location data from SAP S/4HANA on-premise or SAP ECC and to load it to SAP Integrated Business Planning. It uses one of the following Add-Ons for extraction:

- SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_s4hana_addon)
- SAP ERP, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_erp_addon)

For information on the prerequisites of using this template, see SAP ERP or SAP S/4HANA Add-On Integration [page 30].

The `IBP_MD_S4_ERP_AddOn` template uses the interface of the newest support package of the add-ons mentioned above. For information on adaptions that need to be done if you cannot apply the latest support package of the corresponding Add-On, see SAP ERP or SAP S/4HANA Add-On Integration [page 30].

Data Flows of the IBP_MD_S4_ERP_AddOn Task

Target Staging Table in IBP	Data Flow Name	Data Flow Description
SOPMD_STAG_IBPPLANNINGUNIT	IBP_MD_PlanningUnit_AddOn	Writes hard-coded IBP planning unit
SOPMD_STAG_IBPCURRENCY	IBP_MD_Currency_AddOn	Reads currency master data and descriptions from the tables `TCURC` and `TCURT` via the `/IBP/CURRENCY_TEXT` extractor and `SBIC_CURRENCY_TRANSFER` function module and transfers them to the unified planning area `SAPIBP1` as `Currency`
SOPMD_STAG_IBPCUSTOMER	IBP_MD_Customer_AddOn	Reads customer master data from table `/IBP/LOC_EXT` via the `/IBP/LOCATION_ATTR` extractor and transfers it to the unified planning area `SAPIBP1`

Target Staging Table in IBP	Data Flow Name	Data Flow Description
SOPMD_STAG_IBPLOCATION	IBP_MD_Location_AddOn	Reads location master data from table /IBP/LOC_EXT via the /IBP/LOCATION_ATTR extractor and transfers it to the unified planning area SAPIBP1
SOPMD_STAG_IBPPRODUCT	IBP_MD_Product_w_Text_AddOn	Reads product (material) master data and descriptions from the tables /IBP/MAKT_EXT and /IBP/MARA_EXT via the /IBP/PRODUCT_TEXT extractor and transfers them to the unified planning area SAPIBP1
SOPMD_STAG_IBPRESOURCE	IBP_MD_Resource_w_Text_AddOn	Reads resource master data and descriptions from the tables /IBP/RESHEAD_EXT and /IBP/RESDESC_EXT via the /IBP/RESOURCE_ATTR extractor and transfers them to the unified planning area SAPIBP1
SOPMD_STAG_IBPEXCHANGERATE	IBP_MD_ExchangeRate_AddOn	Reads exchange rates from table TCURX via the /IBP/EXCHANGERATES_ATTR extractor and the SBIC_EXRATE_TRANSFER function module and transfers them to the unified planning area SAPIBP1 as Exchange Rate Master Data without time dependency
SOPMD_STAG_IBPLOCATIONPRODUCT	IBP_MD_LoactionProduct_AddOn	Reads location-product (plant-dependent material) master data from table /IBP/MARC_EXT via the /IBP/LOCATIONPRODUCT_ATTR extractor and transfers it to the planning area SAPIBP1

Target Staging Table in IBP	Data Flow Name	Data Flow Description
SOPMD_STAG_IBPRESOURCELOCATI ON	IBP_MD_ResourceLoaction_AddO n	Reads resource-location master data from the tables `/IBP/RESHEAD_EXT` and `/IBP/RESCAPA` (current date only) via the `/IBP/LOCATION_ATTR` extractor and transfers it to the planning area `SAPIBP1`
SOPMD_STAG_IBPPRODUCTIONRESO URCE	IBP_MD_ProductionResource_Ad dOn	Reads production resource master data from table `/IBP/SOSRES_EXT` via the `/IBP/PRODUCTIONRESOURCE_ATTR` extractor and transfers it to the unified planning area `SAPIBP1`
SOPMD_STAG_IBPSOURCEPRODUCTI ON	IBP_MD_SourceProduction_Addo n	Reads production data structure header data from the table `/IBP/SOSHEAD_EXT` via the `/IBP/SOURCEPRODUCTION_ATTR` extractor and transfers it to the unified planning area `SAPIBP1`
SOPMD_STAG_IBPPRODUCTIONSOUR CEITM	IBP_MD_ProductionSourceItm_A ddon	Reads master data of production data structure item data from the tables `/IBP/SOSBOM_EXT`, `/IBP/SOSACT_EXT`, and `/IBP/SOSHEAD_EXT` via the `/IBP/PRODUCTIONSOURCEITEM_ATTR` extractor and transfers it to the unified planning area `SAPIBP1`
SOPMD_STAG_IBPUOMTO	IBP_MD_UOM_w_Text_AddOn	Reads unit of measure master data and descriptions from the tables `T006` and `T006A` via the `/IBP/UOM_TEXT` extractor and `SBIC_MESSURE_TRANSFER` function module and transfers them to the unified planning area `SAPIBP1` as `Target Unit of Measure`
SOPMD_STAG_IBPUOMCONVERSIONF ACTOR	IBP_MD_UOMConversionFactor_A ddOn	Reads master data of product unit of measure conversion factor from the table `/IBP/MARM_EXT` via the `/IBP/PRODUCTUOM_ATTR` extractor and transfers it to the unified planning area `SAPIBP1`

Target Staging Table in IBP	Data Flow Name	Data Flow Description
SOPMD_STAG_IBPSOURCELOCATION	IBP_MD_SourceLocation_AddOn	Reads location source (transportation lane) master data from the table /IBP/TLANE_EXT via the /IBP/SOURCELOCATION_ATTR extractor and transfers it to the unified planning area SAPIBP1
SOPMD_STAG_IBPSOURCECUSTOMER	IBP_MD_SourceCustomer_AddOn	Reads customer source master data via the /IBP/SOURCECUSTOMER_CI_ATTR extractor with customer BAdl implementation and transfers it to the unified planning area SAPIBP1. The /IBP/ETS_SEL_SRCCUST_CI example BAdl implementation reads and aggregates entries from the table /IBP/SLDFCAT_EXT (staging table for sales document field catalog).

How to Use This Template?

1. Import the SAP Integrated Business Planning staging tables you want to use for importing the different types of master data as follows:
   1. Select the datastore that represents your SAP Integrated Business Planning system.
   2. Choose *Import Objects*.
   3. Expand the *Master Data Folder* in the tree.
   4. Select the planning area from which you plan to import the data.
   5. Expand the planning area.
   6. Select the staging tables listed in the table data flows of the IBP_MD_S4_ERP_AddOn task.
   7. Choose *Import*.
2. The name of the staging tables of the master data type are of the pattern SOPMD_STAG_<Master Data Type>. Do not mix them up with the non-staging tables SOPMD_<Master Data Type>.
3. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
4. Create a task in your project to transfer the master data from your SAP S/4HANA or SAP ECC system. Select IBP_MD_S4_ERP_AddOn as template.
5. As source datastore, select the datastore that represents your SAP S/4HANA or SAP ECC system.
6. As target datastore, select the datastore that represents SAP Integrated Business Planning.
7. Choose *Save and Define Data Flow*.
   You can find several target objects (tables) and their data flows under the task you have created. Execute the following steps for each data flow.

8. If you use master data types with a prefix other than `IBP (`, for example, if your location master data type is called `ABCLOCATION` and not `IBPLOCATION)`, replace the target table of the data flow as follows:
   1. Select the data flow.
   2. Choose *Copy to New Target*.
   3. Enter a name for the data flow.
   4. Select the staging table of your master data type as target object. If you have not imported the staging table for your master data type to the datastore yet, you can also perform the import in this step.
   5. Select *Copy Data Flow* or *Import and Copy Data Flow*.
9. Open the data flow for editing by choosing ▶ *Actions* ▶ *Edit* ▶.

> **i Note**
>
> In the note box of the data flow, you can find instructions on adjusting the data flow to fit your needs.

10. Almost every dataflow consists of a source extractor, an IBP target table, and the following intermediate steps:
    1. `QueryAndFilter`: Define filter conditions for the extractor and do a 1:1 mapping of the fields needed for the IBP target table.
    2. `MapSpecialCharsEtc`: Do the mapping of the source field definition to the target field definition, including format changes, replacement of forbidden special characters, and avoiding sending fields with spaces only.
    3. `Map_To_Target`: Mostly 1:1 mapping of the target fields that need to be filled.
       - If required, define additional field mappings for the master data attributes in these steps. The easiest way to achieve this is to start with `QueryAndFilter` and drag and drop the source fields you want to map to the target structure. This automatically adds the field to the input of step `MapSpecialCharsEtc` and you can repeat the same there.
       - You can also define additional filter conditions in step `QueryAndFilter`, but note that not every filter condition is pushed down to the source system. See Pushdown of Filter Conditions To Extractors [page 233] for more details.
11. Once you have adapted each data flow to your needs, set the global variable `$G_LANGUAGE` under *Execution Properties*.
12. Also change the corresponding row of the preload script if `$G_LANGUAGE = '…'` accordingly.
13. Delete the template data flows and their associated target tables from your task.
14. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

There are no prerequisites for this task. Make sure you import the master data before you import the key figure data.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task and choosing ▶ *More Actions* ▶ *Promote* ▶. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

> **i Note**
>
> The data transfer between SAP S/4HANA / SAP ECC and SAP Integrated Business Planning is always a complete data transfer, not a delta transfer.
>
> Master data is only inserted and updated in IBP, but not deleted.

7.2.2 IBP_DDR_MD_ERP_AddOn

The `IBP_DDR_MD_ERP_AddOn` template contains data flows to extract master data from SAP ECC and to load it to the demand-driven replenishment planning area of SAP Integrated Business Planning. It uses SAP ERP, supply chain integration add-on for SAP Integrated Business Planning for extraction (for more information, see http://help.sap.com/ibp_erp_addon).

For information on the prerequisites of using this template, see SAP ERP or SAP S/4HANA Add-On Integration [page 30].

The `IBP_DDR_MD_ERP_AddOn` template uses the interface of the newest support package of the add-ons mentioned above. For information on adaptions that need to be done if you cannot apply the latest support package of the corresponding add-on, see Lifecyle Management for Tasks Based on the IBP Add-On for SAP S/4HANA or SAP ERP [page 34].

Data Flows of the IBP_DDR_MD_ERP_AddOn Task

Target Staging Table in IBP	Data Flow Name	Data Flow Description
`SOPMD_STAG_DDRLOCATION`	`IBP_DDR_MD_Location`	Reads location master data from table `/IBP/LOC_EXT` via the `/IBP/LOCATION_ATTR` extractor and transfers it to the demand-driven replenishment planning area `SAP8`
`SOPMD_STAG_DDRPRODUCT`	`IBP_DDR_MD_Product`	Reads product (material) master data and descriptions from the tables `/IBP/MAKT_EXT` and `/IBP/MARA_EXT` via the `/IBP/PRODUCT_TEXT` extractor and transfers them to the demand-driven replenishment planning area `SAP8`
`SOPMD_STAG_DDRLOCATIONPRODUCT`	`IBP_DDR_MD_LocationProduct`	Reads location-product (plant-dependent material) master data from the table `/IBP/MARC_EXT` via the `/IBP/LOCATIONPRODUCT_ATTR` extractor and transfers it to the demand-driven replenishment planning area `SAP8`

Target Staging Table in IBP	Data Flow Name	Data Flow Description
SOPMD_STAG_IBPSOURCELOCATION	IBP_DDR_MD_SourceLocation	Reads location source (transportation lane) master data from the table /IBP/TLANE_EXT via the /IBP/SOURCELOCATION_ATTR extractor and transfers it to the demand-driven replenishment planning area SAP8
SOPMD_STAG_DDRLOCATIONPRODUCTCURRENCY	IBP_DDR_MD_LocationProductCurrency	Reads location-product (plant-dependent material) master data from the table /IBP/MARC_EXT via the /IBP/LOCATIONPRODUCT_ATTR extractor and transfers it to the demand-driven replenishment planning area SAP8
SOPMD_STAG_DDRPLANNINGUNIT	IBP_DDR_MD_PlanningUnit	Writes hard-coded IBP planning unit
SOPMD_STAG_DDRCURRENCY	IBP_DDR_MD_Currency	Writes hard-coded IBP currency
SOPMD_STAG_DDRCUSTOMER	IBP_DDR_MD_Customer	Writes hard-coded IBP customer with customer group
SOPMD_STAG_DDRBUFFERPROFILE	IBP_DDR_MD_BufferProfile	Writes hard-coded buffer profile table
SOPMD_STAG_DDRLOCATIONPRODUCTCURRENCY	IBP_DDR_MD_LocationProductCurrency	Reads location-product (plant-dependent material) master data from the table /IBP/MARC_EXT via the /IBP/LOCATIONPRODUCT_ATTR extractor and transfers it to the demand-driven replenishment planning area SAP8
SOPMD_STAG_DDRSOURCEPRODUCTION	IBP_DDR_MD_SourceProduction	Reads location-product (plant-dependent material) master data from the table /IBP/SOSHEAD_EXT via the /IBP/SOURCELOCATION_ATTR extractor and transfers it to the demand-driven replenishment planning area SAP8

Target Staging Table in IBP	Data Flow Name	Data Flow Description
SOPMD_STAG_DDRPRODUCTIONSOURCE ITM	IBP_DDR_MD_ProductionSourceItem	Reads data of production data structure item from the tables /IBP/SOSBOM_EXT, /IBP/SOSACT_EXT and /IBP/SOSHEAD_EXT, and eventually /IBP/TLANE_EXT via the /IBP/PRODUCTIONSOURCEITEM_ATTR extractor and transfers it to the demand-driven replenishment planning area SAP8
SOPMD_STAG_DDRSOURCECUSTGROUP	IBP_DDR_MD_SourceCustomerGroup	Reads source customer master data via the /IBP/SOURCECUSTOMER_CI_ATTR extractor with customer BAdI implementation and transfers it to the demand-driven replenishment planning area SAP8. The /IBP/ETS_SEL_SRCCUST_CI example BAdI implementation reads and aggregates entries from the table /IBP/SLDFCAT_EXT (staging table for sales document field catalog). This data flow does an additional aggregation to PRDID and LOCID and adds the hard-coded customer group Customer.

> **i Note**
>
> The IBP_DDR_MD_Supplier2SrcProd dataflow is no longer available, as the supplier product locations now can be read from table /IBP/TLANE_EXT using the /IBP/SOURCELOCATION_ATTR extractor and setting filter condition **ADD_SUPPLIERS = 'X'**. Now each target table is filled by one dataflow, which allows you to set the execution property **Begin post processing** to **After task run is completed** instead of After each data flow, which increases the overall performance of the task.

How to Use This Template?

1. Import the SAP Integrated Business Planning staging tables you want to use for importing the different types of master data as follows:
   1. Select the datastore that represents your SAP Integrated Business Planning system.
   2. Choose *Import Objects*.
   3. Expand the *Master Data Folder* in the tree.
   4. Select the planning area from which you plan to import the data.

5. Expand the planning area.

6. Select the staging tables listed in the table data flows of the `IBP_DDR_MD_ERP_AddOn` task.

7. Choose *Import*.

2. The name of the staging tables of the master data type are of the pattern `SOPMD_STAG_<Master Data Type>`. Do not mix them up with the non-staging tables `SOPMD_<Master Data Type>`.

3. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.

4. Create a task in your project to transfer the master data from your SAP S/4HANA or SAP ECC system. Select `IBP_DDR_MD_ERP_AddOn` as template.

5. As source datastore, select the datastore that represents your SAP S/4HANA or SAP ECC system.

6. As target datastore, select the datastore that represents SAP Integrated Business Planning.

7. Choose *Save and Define Data Flow*.
   You can find several target objects (tables) and their data flows under the task you have created. Execute the following steps for each data flow.

8. If you use master data types with a prefix other than `IBP` (, for example, if your location master data type is called `ABCLOCATION` and not `IBPLOCATION)`, replace the target table of the data flow as follows:

   1. Select the data flow.

   2. Choose *Copy to New Target*.

   3. Enter a name for the data flow.

   4. Select the staging table of your master data type as target object. If you have not imported the staging table for your master data type to the datastore yet, you can also perform the import in this step.

   5. Select *Copy Data Flow* or *Import and Copy Data Flow*.

9. Open the data flow for editing by choosing ▌ *Actions* ▶ *Edit* ▌.

> **ⓘ Note**
>
> In the note box of the data flow, you can find instructions on adjusting the data flow to fit your needs.

10. Almost every dataflow consists of a source extractor, an IBP target table, and the following intermediate steps:

    1. `QueryAndFilter`: Define filter conditions for the extractor and do a 1:1 mapping of the fields needed for the IBP target table.

    2. `MapSpecialCharsEtc`: Do the mapping of the source field definition to the target field definition, including format changes, replacement of forbidden special characters, and avoiding sending fields with spaces only.

    3. `Map_To_Target`: Mostly 1:1 mapping of the target fields that need to be filled.

       - If required, define additional field mappings for the master data attributes in these steps. The easiest way to achieve this is to start with `QueryAndFilter` and drag and drop the source fields you want to map to the target structure. This automatically adds the field to the input of step `MapSpecialCharsEtc` and you can repeat the same there.

       - You can also define additional filter conditions in step `QueryAndFilter`, but note that not every filter condition is pushed down to the source system. See Pushdown of Filter Conditions To Extractors [page 233] for more details.

11. Once you have adapted each data flow to your needs, navigate to the *Execution Properties* and set the following global variables according to your needs:

    - `$G_LANGUAGE`: needs to be set to the selection language for descriptions

    - `$G_TARGET_CURRENCY`: needs to be set to the currency used in IBP to store the cost per unit

- $G_EXCHANGERATETYPE: needs to be set to the exchange rate type for converting the cost per unit to the IBP currency
- $G_EXCHANGERATEDATE: needs to be set to the exchange rate date for converting the cost per unit to the IBP currency

12. Also change the corresponding row of the preload script accordingly:
    - if $G_LNGUAGE = '...' (replace ... with your newly set value)
    - if $G_TARGET_CURRENCY = '...' (replace ... with your newly set value)
    - if $G_EXCHANGERATETYPE = '...' (replace ... with your newly set value)
    - if $G_EXCHANGERATECURRENCY = '...' (replace ... with your newly set value)

13. Delete the template data flows and their associated target tables from your task.

14. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

There are no prerequisites for this task. Make sure you import the master data before you import the key figure data.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task and choosing ▶ *More Actions* ▶ *Promote* ▶. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

> **i Note**
>
> The data transfer between SAP S/4HANA / SAP ECC and SAP Integrated Business Planning is always a complete data transfer, not a delta transfer.
>
> Master data is only inserted and updated in IBP, but not deleted.

7.2.3 IBP_DDR_OBP_KF_to_File

You can use the IBP_DDR_OBP_KF_to_File template together with IBP_DDR_OBP_MD_from_File as a starting point to transfer demand-driven-replenishment-relevant master data attributes and key figures that are required for order-based planning (OBP). As some of the key figures in demand-driven replenishment have a static pattern, these key figures are stored as master data attributes.

For the data transfer from your demand-driven replenishment planning area to your OBP planning area, two data flows are provided as templates, which you can use in SAP Cloud Integration for data services. The IBP_DDR_OBP_KF_to_File data flow exports the relevant data into a CSV file, while IBP_DDR_OBP_MD_from_File updates OBP master data attributes with this data.

> **i Note**
>
> The data flow contains a delta filter logic for decoupling points, which compares data to the values from yesterday. In case there is a change in decoupling points within the same day, it is not detected, and the

starting value of the day remains in the system. If this is a valid use case for your business, you need to further enhance the template.

The template consists of the following data flow:

Data Flow of IBP_DDR_OBP_KF_to_File

Data Flow Name	Description	Source Fields
IBP_DDR_OBP_KF_to_File	Write relevant demand-driven replenishment key figures and master data attributes to a file to integrate them into your OBP planning area as master data attributes	PRDID
		LOCID
		MRPTYPE
		DDOSTPCT
		RECDECOUPPOINTIND
		DDDECOUPPPOINTCHANGEINDICATOR
		DECOUPLEDLEADTIMEDAYS
		DDORDERSPIKEHORIZON
		DDADJUSTEDADU
		PERIODID0
		TSTFR

Prerequisites for Creating a Task Based on the IBP_DDR_OBP_KF_to_File Template

Creating Datastores

1. If not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading demand-driven replenishment data from the planning area in SAP IBP:
   1. Select the datastore that represents your SAP IBP system and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

   > **i Note**
   >
   > The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`. Abbreviations may occur in the name of the calculation scenario.

3. In SAP Cloud Integration for data services, create a datastore of the type *File Format Group*.
   Provide the following data for the new datastore:

Data Type	Value
Name	A name of your choice
Description	A description of your choice
Type	*File Format Group*
Agent	The SAP Data Services Agent in your landscape
Location	At Agent
Root Directory	A directory where your agent can store data

For all other fields, use the default values.

4. Create the file format as follows:
   1. Navigate to the *File Formats* tab.
   2. Choose *Create File Format*.
   3. Choose *Create From Scratch*.
      Provide the following data for the new datastore:

Data Type	Value
Name	A name of your choice, for example, `Master_Data_OBP`
Description	A description of your choice
First row contains column names	X

For all other fields, use the default values.

4. Select your newly created file format and navigate to *Columns*. Choose *Add Column* and provide the following data:

Data Type	Value	Description
LOCID	varchar(20)	Location ID
PRDID	varchar(40)	Product ID
AVERAGEDAILYUSAGE	decimal(18,6)	Average Daily Usage
DECOUPLEDLEADTIME	integer	Decouple Lead Time
ORDERSPIKEHORIZON	integer	Order Spike Horizon
ORDERSPIKETHRESHOLDPERCENT	decimal(18,6)	Order Spike Threshold Percentage
DECOUPLINGPOINTDEL	varchar(1)	Decoupling Point Marked for Del.

How to Use This Template?

1. As source datastore, select the datastore that represents your SAP IBP system. As target datastore, select the datastore of the type *File Format Group* that you have previously created. Choose *Save and Define Data Flow*.
2. Select `IBP_DDR_OBP_KF_to_File` as template.
3. Open the data flow for editing by choosing ❚▶ *Actions* ❯ *Edit* ❚.

> **i Note**
>
> In the note box of the data flow, you can find instructions on adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

4. Replace the source table with your own calculation view and recreate the mapping arrow from the source table to the `QueryAndFilter` transform.
5. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

> **i Note**
>
> Make sure that the `IBP_DDR_OBP_KF_to_File` and `IBP_DDR_OBP_MD_from_File` data flows are executed together. For more information, see IBP_DDR_OBP_MD_from_File [page 49].

7.2.4 IBP_DDR_OBP_MD_from_File

You can use the `IBP_DDR_OBP_MD_from_File` template together with `IBP_DDR_OBP_KF_to_File` as a starting point to transfer demand-driven-replenishment-relevant master data attributes and key figures that are required for order-based planning (OBP). As some of the key figures in demand-driven replenishment have a static pattern, these key figures are stored as master data attributes.

For the data transfer from your demand-driven replenishment planning area to your OBP planning area, two data flows are provided as templates, which you can use in SAP Cloud Integration for data services. The `IBP_DDR_OBP_KF_to_File` data flow exports the relevant data into a CSV file, while `IBP_DDR_OBP_MD_from_File` updates OBP master data attributes with this data.

The template consists of the following data flow:

Data Flow of IBP_DDR_OBP_MD_from_File

Target Staging Table in SAP IBP	Data Flow Name	Data Flow Description
SOPMD_STAG_S7DDRDECOUPLINGPOINT	IBP_DDR_OBP_MD_from_File	Read data from CSV file and transfer it to your OBP planning area (sample planning area SAP7)

How to Use This Template?

1. Import the SAP IBP staging tables that you want to use for importing the master data type as follows:
   1. Select the datastore that represents your SAP IBP system.
   2. Choose *Import Objects*.
   3. Expand *Master Data Folder* in the tree.
   4. Select the planning area from which you want to import the data.
   5. Expand the planning area, and select the staging table `SOPMD_STAG_XXXDDRDECOUPLINGPOINT`, where **XXX** is the prefix of your master data type.
   6. Choose *Import*.

   > **i Note**
   >
   > The names of the staging tables for master data types follow the pattern `SOPMD_STAG_<Master Data Type>`. Do not mix them up with the non-staging tables `SOPMD_<Master Data Type>`.

2. Under *Projects*, create a project in which you can group your data integration tasks, or select an existing project.
3. Create a task in your project to transfer the master data from the CSV file. Select `IBP_DDR_OBP_MD_from_File` as template.
4. As source datastore, select the file format group datastore you created for the `IBP_DDR_OBP_KF_to_File` data flow.
5. As target datastore, select the datastore that represents your SAP IBP system.
6. Choose *Save and Define Data Flow*.

> **i Note**
>
> You can find a target object (tables) and a data flow under the task you have created.

7. If you use master data types with a prefix other than S7 (for example, if your location master data type is called **ABCDDRDECOUPLINGPOINT** and not **S7DDRDECOUPLINGPOINT**), replace the target table of the data flow as follows:
   1. Select the data flow, and choose *Copy to New Target*.
   2. Enter a name for the data flow.
   3. Select the staging table of your master data type as the target object. If you have not imported the staging table for your master data type to the datastore yet, you can do so now.
   4. Select *Copy Data Flow* or *Import and Copy Data Flow*.
8. Open the data flow for editing by choosing ▐ *Actions* ❯ *Edit* ▐.

> **i Note**
>
> In the note box of the data flow, you can find instructions on adjusting the data flow to fit your needs, for example, by replacing the CSV data source.

9. Once you have adapted the data flow to your needs, navigate to *Execution Properties* and set the global variables according to your needs.
10. Delete the template data flows and their associated target tables from your task.
11. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

> **i Note**
>
> The data transfer between the CSV file and SAP IBP is always a complete data transfer, not a delta load.
>
> Master data is only inserted and updated in SAP IBP, but not deleted.

> **i Note**
>
> Make sure that the `IBP_DDR_OBP_KF_to_File` and `IBP_DDR_OBP_MD_from_File` data flows are executed together. For more information, see IBP_DDR_OBP_KF_to_File [page 45].
>
> After a successful test of the two data flows `IBP_DDR_OBP_KF_to_File` and `IBP_DDR_OBP_MD_from_File`, we recommend to combine them into a process flow so that they are executed together.

Working with Process Flows

Creating a Process Flow

You can create a process flow as follows:

1. Select your project and choose *Create Process*.
2. Enter a name and description for your process, such as **Process_Flow_IBP_DDR_OBP_via_File**.
3. Select the checkbox *Load to SAP Integrated Business Planning (requires post-processing)*.
4. As target datastore, select the datastore that represents your SAP IBP system.
5. Choose *Save and Edit Process*.
6. Add your first data flow `IBP_DDR_OBP_KF_to_File`.
7. Add a script to the process flow and name it **PreloadScript**. Then, connect the script node with the `IBP_DDR_OBP_KF_to_File` data flow.
8. Add your second data flow `IBP_DDR_OBP_MD_from_File`. Connect it with the **PreloadScript** node.
9. Navigate to the `IBP_DDR_OBP_MD_from_File` data flow and choose *Execution Properties*. Copy the script under *Preload*.
10. Navigate back to the process flow and open the **PreloadScript**. Replace the sample script with the copied one.
11. Add another script to the process flow and name it **PostloadScript**. Then, connect the script node with the `IBP_DDR_OBP_MD_from_File` data flow.
12. Navigate to the `IBP_DDR_OBP_MD_from_File` data flow and choose *Execution Properties*. Copy the script under *Postload*.
13. Navigate back to the process flow and open the **PostloadScript**. Replace the sample script with the copied one.

Executing the Process Flow

To execute a process flow, go to the *Projects* view, and choose *Run Now*. You can check the process flow run (view a log of errors and the number of records transferred) by selecting your process flow, and choosing *View History*.

After a thorough testing of your process flow, you may decide to schedule it for a periodic run. To do so, first promote your process flow to the production environment by selecting it and choosing ▶ *More Actions* ▶ *Promote* ▶. Then, switch to the productive view. There, select your process flow, and choose *Schedule*, then *New*, and enter the required data.

7.2.5 SOP_MD_LocationMaster

This is a task for extracting **location** master data from SAP ERP, and loading it to SAP Integrated Business Planning.

Source table is `T001W`, source extractor is `0PLANT_ATTR`.

The task loads master data into the staging table of the `SM1LOCATION` master data type (`SOPMD_STAG_SM1LOCATION`).

Prerequisites for Creating a Task Based on the SOP_MD_LocationMaster Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the location master data from SAP ERP. Select *Use Template*, and select SOP_MD_LocationMaster as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.2.6 SOP_MD_ProductMaster

This is a task for extracting **product** master data from SAP ERP, and loading it to SAP Integrated Business Planning.

Source extractor is 0MATERIAL_ATTR, source table is MAKT.

The task loads master data into the staging table of the SM1PRODUCT master data type (SOPMD_STAG_SM1PRODUCT).

Prerequisites for Creating a Task Based on the SOP_MD_ProductMaster Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the product master data from SAP ERP. Select *Use Template*, and select SOP_MD_ProductMaster as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.

5. Review the default values of the global variables, and make changes as needed.

7.2.7 SOP_MD_LocationProd

This is a task for extracting **location product** master data from SAP ERP, and loading it to SAP Integrated Business Planning.

Source tables are MBEW, MARA, and MARC.

The task loads master data into the staging table of the SM1LOCATIONPRODUCT master data type (SOPMD_STAG_SM1LOCATIONPRODUCT).

Prerequisites for Creating a Task Based on the SOP_MD_LocationMaster Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the location product master data from SAP ERP. Select *Use Template*, and select SOP_MD_LocationProd as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.2.8 SOP_MD_CustomerMaster

This is a task for extracting **customer** master data from SAP ERP, and loading it to SAP Integrated Business Planning.

Source extractor is 0CUSTOMER_ATTR, source tables are KNVP, and KNVH.

The task loads master data into the staging table of the SM1CUSTOMER master data type (SOPMD_STAG_SM1CUSTOMER).

Prerequisites for Creating a Task Based on the SOP_MD_CustomerMaster Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the customer master data from SAP ERP. Select *Use Template*, and select SOP_MD_CustomerMaster as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.2.9 SOP_MD_Resource

This is a task for extracting **resource** master data from SAP APO, and loading it to SAP Integrated Business Planning.

Source table is /SAPAPO/RES_HEAD.

The task loads master data into the staging table of the SM1RESOURCE master data type (SOPMD_STAG_SM1RESOURCE).

Prerequisites for Creating a Task Based on the SOP_MD_Resource Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the resource master data from SAP ERP. Select *Use Template*, and select SOP_MD_Resource as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.2.10 IBP_MD_PlanningArea

The `IBP_MD_PlanningArea` template contains five data flows to extract product master data, unit of measure, unit of measure conversion factors, location master data, and customer master data from SAP APO, and load it to SAP Integrated Business Planning.

Data Flows of the IBP_MD_PlanningArea Task

Target Object	Data Flow Name	Data Flow Description
`SOPMD_STAG_IBPPRODUCT` staging table	`DF_IBP_ProductMaster`	`DF_IBP_ProductMaster` - Extract product master data from a planning area DataSource in SAP SCM/SAP APO. Texts are read from the text table of the product InfoObject.
`SOPMD_STAG_IBPUOMTO` staging table	`DF_IBP_UnitsOfMeasure`	`DF_IBP_UnitsOfMeasure` - Extract all units of measure that are defined for the products in the planning area.
`SOPMD_STAG_IBPUOMCONVERSIONFACTOR` staging table	`DF_IBP_UnitsOfMeasure_Conversion`	`DF_IBP_UnitsOfMeasure_Conversion` - Extract the conversion factors for all units of measure that are defined for the products in the planning area. The factors refer to conversion from the base unit of measure into any of the other units of measure.
`SOPMD_STAG_IBPLOCATION` staging table	`DF_IBP_LocationMaster`	`DF_IBP_LocationMaster` - Extract location master data from a planning area DataSource in SAP SCM/SAP APO.
`SOPMD_STAG_IBPCUSTOMER` staging table	`DF_IBP_CustomerMaster`	`DF_IBP_CustomerMaster` - Extract customer master data from a planning area DataSource in SAP SCM/SAP APO.

Prerequisites for Creating a Task Based on the IBP_MD_PlanningArea Template

Defining and Releasing DataSources in SAP APO

In SAP APO, you have executed the following steps:

1. You have set up your planning area in SAP APO Demand Planning.
2. You have generated the necessary DataSources (in the *DP/SNP Data Extraction* (`/SAPAPO/SDP_EXTR`) transaction).

To be able to upload data to the demand application, you need to extract the following master data types: **product**, **location**, and **customer**. Generate a DataSource for each master data type that you want to upload from SAP APO.

For extracting master data, select the relevant ID field for the master data type, the planning version, and the attributes you want to extract, if any.

Make sure you have hidden all fields that you do not need to extract.

3. Replicate each DataSource.
4. In the *Data Warehousing Workbench*, activate the DataSources.
5. Execute the *Release 9A DataSources for External Use* (/SAPAPO/PAREA_EXTR_EXPOSE) report to release your DataSources for external use, that is, to make them available for SAP Cloud Integration for data services.

> i Note
>
> In case the /SAPAPO/PAREA_EXTR_EXPOSE report is not available in your system, implement SAP Note 2085981 .

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

Prerequisite: SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP APO system. Provide the following data for the new datastore:

Name and description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP APO / SAP SCM system	

2. Import the DataSources that you have created in SAP APO. Select the datastore that represents your SAP APO / SAP SCM system, and choose *Import Object by Name*. Select *Extractors*, and enter the name of the DataSource in the *Name* field.
3. Create a datastore that represents your SAP Integrated Business Planning system. Provide the following data for the new datastore:

Name and description	
Type	Integrated Business Planning

> i Note
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

4. Import the SAP Integrated Business Planning staging tables that will be used for importing **product, location and customer master data, and unit of measure data**:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Master Data Folder* in the tree, and select the planning area from which you will import the data.
   3. Expand the planning area, select the staging tables of the product, location and customer master data types, of the units of measure and their conversion, and choose *Import*.

   > i Note
   >
   > The name of the staging table of the master data type is of the following pattern:
   > `SAPSOPG.SOPMD_STAG_<Master Data Type>`.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the product, unit of measure, location, and customer master data from your planning area in SAP APO. Select *Use Template*, and select `IBP_MD_PlanningArea` as template.
3. As source datastore, select the datastore that represents your SAP APO / SAP SCM system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose Save and Define Data Flow.
   You can find five target objects (tables) and their data flows under the task you have created.
   Execute the following steps for each data flow.
4. If you use different master data types than the ones in the template (`IBPPRODUCT`, `IBPUOMTO`, `IBPUOMCONVERSION`, `IBPLOCATION`, `IBPCUSTOMER`), replace the target table of the data flow. To do this, select the data flow, and choose *Copy to new target*.
5. Enter a name for the data flow, and select the staging table of your master data type as target object. If you have not imported the staging table for your master data type to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Open the data flow for editing by choosing ▶ *Actions* ❯ *Edit.* ▶.

   > i Note
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

7. Replace the source master data table (`9ADP_CUSTOM_PRODUCT`, `9ADP_CUSTOM_LOCATION`, or `9ADP_CUSTOM_CUSTOMER`) with your DataSource of the same master data type (the DataSource you have generated and exposed in SAP APO).

8. Depending on your data flow, connect the source table to the `Read_Products`, `Read_Locations`, or `Read_Customers` step, and define field mappings for the master data, for the attributes (if available), and for the planning version.

9. If applicable, replace the source text table for the master data type with the text table for your InfoObject of the same master data type.
   To get the text table of your InfoObject, go to *Edit InfoObjects* (`RSD1`) transaction, select your InfoObject, and choose *Display*. Go to the *Master Data/Texts* tab to find out the name of the source text table.

10. Connect the source text table to the `Read_Texts` step, and define field mappings for the master data, description, and language.

11. Do not replace the `/SAPAPO/MATKEY` and `T006` tables, and do not change their mappings.

12. If required, define additional field mappings for the master data attributes in the `MapToTarget` step. Make sure you have all key fields mapped.

13. Once you have adjusted each data flow to your needs, set the following global variables under Execution Properties:
    - `$G_PLANNING_VERSION`
    - `$G_LANGUAGE`

14. Delete the template data flows and their associated target tables from your task.

15. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

There are no prerequisites for this task. Make sure you import the master data before you import the key figure data.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ❯ *Promote* ◀. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

> **i Note**
>
> Between SAP APO and SAP Integrated Business Planning, the data transfer is always a complete data transfer, not a delta transfer.
>
> Deletions of characteristic value combinations in SAP APO's Demand Planning cannot be transferred to SAP Integrated Business Planning.

7.2.11 IBP_MD_SalesOrder_InitialLoad

The `IBP_MD_SalesOrder_InitialLoad` template contains a data flow to extract sales orders from SAP ERP, and load it to SAP Integrated Business Planning for demand.

The `IBP_MD_SalesOrder_InitialLoad` template transfers the current status of the sales orders that have been created up to a certain point in time.

Prerequisites for Creating a Task Based on the IBP_MD_SalesOrder_InitialLoad Template

Setting Up a Planning Area in SAP Integrated Business Planning

You have a planning area that contains the sales order as a master data type, such as the `SAP6` sample planning area.

In the `SAP6` sample planning area, the following master data type, planning levels, and key figures are defined for sales orders:

Master Data Type	IBPSALESORDER
Planning Levels	`SALESORDERDAILY`, `SALESORDERUOMTODAILY`
Key Figures	`REQQTY` (Requested Quantity), `CONFQTY` (Confirmed Quantity)

The `REQQTY` and `CONFQTY` attributes of the sales order master data type are assigned as *Attributes as Key Figures* to the planning area.

Activating DataSources in SAP ERP

In SAP ERP, you have executed the following steps:

1. You have activated the `2LIS_11_VASCL` DataSource (in the *LO Data Extraction: Customizing Cockpit* (`LBWE`) transaction).
2. Delete the BW setup data. Execute the *Delete Newly Reorg. BW Data* (`LBWG`) transaction for application 11 (*SD Sales BW*).
3. Rebuild the setup tables. To do so, start the *Reorg. of VIS Extr. Struct.: Order* (`OLI7BW`) transaction. Enter a name for the run, and a time that is ahead enough in the future that the job finishes. For more information, see the documentation of the `OLI7BW`) transaction.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

Prerequisite: SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

1. In case you don't have a datastore in SAP Cloud Integration for data services that represents your SAP ERP system, create one. Provide the following data for the new datastore:

Name and Description

Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP ERP system	

2. Import the DataSource that you have activated in SAP ERP. Select the datastore that represents your SAP ERP system, and choose *Import Object by Name*. Select *Extractors*, and enter `2LIS_11_VASCL` in the *Name* field. Select *Change Data Capture* for the *Extraction Mode*, and enter a freely defined value in the *Subscriber* field.

   > i Note
   >
   > As several systems may read the same DataSource, the subscriber ID is used to differentiate which changes have been already read by a specific system. When reading data from a DataSource in SAP ERP in delta mode, the subscriber ID is passed to the DataSource. The delta queue determines which data has already been read by this subscriber and returns the unread delta only.
   >
   > The template uses `IBP` as the default subscriber ID. If you also use `IBP`, you don't have to replace the DataSource when you create your own task based on this template.

3. In case you don't have a datastore in SAP Cloud Integration for data services that represents your SAP Integrated Planning system, create one.
   Provide the following data for the new datastore:

   Name and Description

Type	Integrated Business Planning
Instance	Your SAP Integrated Business Planning system

   > i Note
   >
   > If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

4. Import the SAP Integrated Business Planning staging tables that will be used for importing sales order data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Master Data Folder* in the tree, and select the planning area from which you will import the data.
   3. Expand the planning area, select the staging table for the sales order, and choose *Import*.

      > i Note
      >
      > The name of the staging table is of the following pattern: `SOPMD_STAG_<Master Data Type>`.

> Abbreviations may occur in the name of the table.
>
> Make sure to select a table that matches this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the sales order data from SAP ERP. Select *Use Template*, and select `IBP_MD_SalesOrder_InitialLoad` as template.
3. As source datastore, select the datastore that represents your SAP ERP system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*.
   You can find the `SOPMD_STAG_IBPSALESORDER` table and the `DF_IBP_SalesOrder_InitialLoad` data flow under the task you have created.
4. If you use different a master data type for sales orders than the one in the template (`IBPSALESORDER`), replace the target table of the data flow. To do this, select the data flow, and choose *Copy to new target*.
5. Enter a name for the data flow, and select the staging table of your sales order master data type as target object. If you have not imported the staging table for your master data type to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Open the data flow for editing by choosing ▐ *Actions* ▶ *Edit.* ▐.

   > **i Note**
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

   If you use `IBP` as the subscriber ID, skip steps 7 and 8.
7. In case the subscriber ID you have defined is different from `IBP`, replace the DataSource (`2LIS_11_VASCL_IBP`) with the DataSource that has - as suffix - the subscriber ID you defined.
8. Connect your DataSource to the `Read_Source` step, and define field mappings for the fields of the output structure.
9. In the `Read_Source` step, select a field on the left side, and go to the *Extractor Options* tab. Make sure that *Initial Load* is set to *Yes*. This is to distinguish between initial load and delta load.
10. Choose *Close*, then *Done* to exit the data flow editor.
11. Delete the template data flows and their associated target tables from your task.
12. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Product, location, and customer master data that is referenced by the sales orders must exist in SAP Integrated Business Planning before you execute the task to load sales orders.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

7.2.12 IBP_MD_SalesOrder_DeltaLoad

The `IBP_MD_SalesOrder_DeltaLoad` template contains a data flow to extract sales orders from SAP ERP, and load it to SAP Integrated Business Planning for demand.

The `IBP_MD_SalesOrder_DeltaLoad` template transfers sales orders that have been created or changed since the last data transfer. For sales orders that have been deleted in SAP ERP, the values in the requested quantity and confirmed quantity fields are set 0 in SAP Integrated Business Planning, but the sales order itself stays in the system.

Prerequisites for Creating a Task Based on the IBP_MD_SalesOrder_DeltaLoad Template

The prerequisites are the same with those of the `IBP_MD_SalesOrder_InitialLoad` task. Use the same datastores, target table, and subscriber ID.

However, **for the delta load of sales orders, don't delete and rebuild BW setup data** (don't use the `LBWG` and `OLI7BW` transactions).

In the *LO Data Extraction: Customizing Cockpit* (`LBWE`) transaction, schedule a periodic job for application **11** (*SD Sales BW*). The job you define here should complete before the job for sales order delta load in SAP Cloud Integration for data services starts.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the sales order data from SAP ERP. Select *Use Template*, and select `IBP_MD_SalesOrder_DeltaLoad` as template.
3. As source datastore, select the datastore that represents your SAP ERP system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*.
   You can find the `SOPMD_STAG_IBPSALESORDER` table and the `DF_IBP_SalesOrder_DeltaLoad` data flow under the task you have created.
4. If you use different a master data type for sales orders than the one in the template (`IBPSALESORDER`), replace the target table of the data flow. To do this, select the data flow, and choose *Copy to new target*.
5. Enter a name for the data flow, and select the staging table of your sales order master data type as target object. Select *Copy Data Flow*.
6. Open the data flow for editing by choosing ▌▶ *Actions* ❯ *Edit.* ▐.

   > **i** Note
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

   If you use `IBP` as the subscriber ID, skip steps 7 and 8.

7. In case the subscriber ID you have defined is different from `IBP`, replace the DataSource (`2LIS_11_VASCL_IBP`) with the DataSource that has - as suffix - the subscriber ID you defined.

8. Connect your DataSource to the `Read_Source` step, and define field mappings for the fields of the output structure.

9. In the `Read_Source` step, select a field on the left side, and go to the *Extractor Options* tab. Make sure that *Initial Load* is set to *No*. This is to distinguish between initial load and delta load.

10. Choose *Close*, then *Done* to exit the data flow editor.

11. Delete the template data flows and their associated target tables from your task.

12. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Product, location, and customer master data that is referenced by the sales orders must exist in SAP Integrated Business Planning before you execute the task to load sales orders.

You have executed the task for the initial load of sales order data.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, we recommend that you schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▌ *More Actions* ▸ *Promote* ▌. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

Before this job runs, run a job in SAP ERP, in transaction `LBWE` for application **11**, to build up the delta queue of sales order data to be transferred. For more information, see the Prerequisites section for this template.

7.2.13 IBP_MD_Deliveries_InitialLoad

The `IBP_MD_Deliveries_InitialLoad` template contains a data flow to extract outbound deliveries from SAP ERP, and load it to SAP Integrated Business Planning for demand.

The `IBP_MD_Deliveries_InitialLoad` template transfers the current status of outbound deliveries that have been created up to a certain point in time.

Prerequisites for Creating a Task Based on the IBP_MD_Deliveries_InitialLoad Template

Setting Up a Planning Area in SAP Integrated Business Planning

You have a planning area that contains the outbound deliveries as a master data type, such as the `SAP6` sample planning area.

In the SAP6 sample planning area, the following master data type, planning levels, and key figures are defined for outbound deliveries:

Master Data Type	IBPDELIVERY
Planning Levels	DELIVERYDAILY, DELIVERYUOMTODAILY
Key Figures	DELIVQTY (Delivered Quantity)

The DELIVQTY attribute of the outbound delivery master data type is assigned as *Attributes as Key Figures* to the planning area.

Activating DataSources in SAP ERP

In SAP ERP, you have executed the following steps:

1. You have activated the 2LIS_12_VCITM DataSource (in the *LO Data Extraction: Customizing Cockpit* (LBWE) transaction).
2. Delete the BW setup data. Execute the *Delete Newly Reorg. BW Data* (LBWG) transaction for application 12 (*LE Shipping BW*).
3. Rebuild the setup tables. To do so, start the *Reorg. VIS Extr. Str.: Delivery* (OLI8BW) transaction. Enter a name for the run, and a time that is ahead enough in the future that the job finishes. For more information, see the documentation of the OLI8BW) transaction.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

Prerequisite: SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

1. Use the same datastore - representing your SAP ERP system - that you use for transferring sales order data from SAP ERP.
2. Import the DataSource that you have activated in SAP ERP. Select the datastore that represents your SAP ERP system, and choose *Import Object by Name*. Select *Extractors*, and enter 2LIS_12_VCITM in the *Name* field. Select *Change Data Capture* for the *Extraction Mode*. SAP recommends that you enter the same value in the *Subscriber* field that you entered when you exported the sales orders.

> **i Note**
>
> As several systems may read the same DataSource, the subscriber ID is used to differentiate which changes have been already read by a specific system. When reading data from a DataSource in SAP ERP in delta mode, the subscriber ID is passed to the DataSource. The delta queue determines which data has already been read by this subscriber and returns the unread delta only.
>
> The template uses IBP as the default subscriber ID. If you also use IBP, you don't have to replace the DataSource when you create your own task based on this template.

3. Use the same datastore - representing your SAP Integrated Business Planning system - that you use for transferring sales order data from SAP ERP.

4. Import the SAP Integrated Business Planning staging tables that will be used for importing outbound deliveries data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Master Data Folder* in the tree, and select the planning area from which you will import the data.
   3. Expand the planning area, select the staging table for the outbound deliveries, and choose *Import*.

   > **i Note**
   >
   > The name of the staging table is of the following pattern: `SOPMD_STAG_<Master Data Type>`.
   >
   > Abbreviations may occur in the name of the table.
   >
   > Make sure to select a table that matches this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the outbound deliveries data from SAP ERP. Select *Use Template*, and select `IBP_MD_Deliveries_InitialLoad` as template.
3. As source datastore, select the datastore that represents your SAP ERP system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*.
   You can find the `SOPMD_STAG_IBPDELIVERY` table and the `DF_IBP_Deliveries_InitialLoad` data flow under the task you have created.
4. If you use different a master data type for outbound deliveries than the one in the template (`IBPDELIVERY`), replace the target table of the data flow. To do this, select the data flow, and choose *Copy to new target*.
5. Enter a name for the data flow, and select the staging table of your outbound delivery master data type as target object. If you have not imported the staging table for your master data type to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Open the data flow for editing by choosing ▌▶ *Actions* ▶ *Edit.* ▌.

   > **i Note**
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

   If you use `IBP` as the subscriber ID, skip steps 7 and 8.
7. In case the subscriber ID you have defined is different from `IBP`, replace the DataSource (`2LIS_12_VCITM_IBP`) with the DataSource that has - as suffix - the subscriber ID you defined.
8. Connect your DataSource to the `Read_Source` step, and define field mappings for the fields of the output structure.
9. In the `Read_Source` step, select a field on the left side, and go to the *Extractor Options* tab. Make sure that *Initial Load* is set to *Yes*. This is to distinguish between initial load and delta load.
10. Choose *Close*, then *Done* to exit the data flow editor.
11. Delete the template data flows and their associated target tables from your task.

12. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Product, location, and customer master data that is referenced by the outbound deliveries must exist in SAP Integrated Business Planning before you execute the task to load outbound deliveries.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

7.2.14 IBP_MD_Deliveries_DeltaLoad

The `IBP_MD_Deliveries_DeltaLoad` template contains a data flow to extract outbound deliveries from SAP ERP, and load it to SAP Integrated Business Planning for demand.

The `IBP_MD_Deliveries_DeltaLoad` template transfers outbound deliveries that have been created or changed since the last data transfer. For outbound deliveries that have been deleted in SAP ERP, the value in the delivered quantity field is set 0 in SAP Integrated Business Planning, but the outbound delivery itself stays in the system.

Prerequisites for Creating a Task Based on the IBP_MD_Deliveries_DeltaLoad Template

The prerequisites are the same with those of the `IBP_MD_Deliveries_InitialLoad` task. Use the same datastores, target table, and subscriber ID.

However, **for the delta load of outbound deliveries, don't delete and rebuild BW setup data** (don't use the `LBWG` and `OLI8BW` transactions).

In the *LO Data Extraction: Customizing Cockpit* (`LBWE`) transaction, schedule a periodic job for application **12** (*LE Shipping BW*). The job you define here should complete before the job for outbound deliveries delta load in SAP Cloud Integration for data services starts.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the outbound deliveries from SAP ERP. Select *Use Template*, and select `IBP_MD_Deliveries_DeltaLoad` as template.
3. As source datastore, select the datastore that represents your SAP ERP system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*.
   You can find the `SOPMD_STAG_IBPDELIVERY` table and the `DF_IBP_Deliveries_DeltaLoad` data flow under the task you have created.

4. If you use different a master data type for sales orders than the one in the template (`IBPDELIVERY`), replace the target table of the data flow. To do this, select the data flow, and choose *Copy to new target*.

5. Enter a name for the data flow, and select the staging table of your outbound delivery master data type as target object. Select *Copy Data Flow*.

6. Open the data flow for editing by choosing ▶ *Actions* ❯ *Edit.* ❩.

> **i Note**
>
> In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

If you use `IBP` as the subscriber ID, skip steps 7 and 8.

7. In case the subscriber ID you have defined is different from `IBP`, replace the DataSource (`2LIS_12_VCITM_IBP`) with the DataSource that has - as suffix - the subscriber ID you defined.

8. Connect your DataSource to the `Read_Source` step, and define field mappings for the fields of the output structure.

9. In the `Read_Source` step, select a field on the left side, and go to the *Extractor Options* tab. Make sure that *Initial Load* is set to *No*. This is to distinguish between initial load and delta load.

10. Choose *Close*, then *Done* to exit the data flow editor.

11. Delete the template data flows and their associated target tables from your task.

12. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Product, location, and customer master data that is referenced by the outbound deliveries must exist in SAP Integrated Business Planning before you execute the task to load outbound deliveries.

You have executed the task for the initial load of outbound deliveries data.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, we recommend that you schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ❯ *Promote* ❩. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

Before this job runs, run a job in SAP ERP, in transaction `LBWE` for application **12**, to build up the delta queue of outbound delivery data to be transferred. For more information, see the Prerequisites section for this template.

7.2.15 `IBP_MRO_SAP9_MD_PM_S4_ERP_AddOn`

The `The IBP_MRO_SAP9_MD_PM_S4_ERP_AddOn` template contains data flows to extract MRO spare parts, maintenance plans, location, resource, equipment, functional location, location product, plant-dependent maintenance data, purchasing information records, transportation lanes, resource location, resource time consumption and product consumption data from the task list, currency, exchange rate data from SAP S/

4HANA on-premise or SAP ECC, and to load it to a planning area based on the SAP9 sample planning area in SAP Integrated Business Planning for Supply Chain (SAP IBP).

It uses one of the following add-ons for extraction:

- SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_s4hana_addon)
- SAP ERP, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_erp_addon)

Data Flows of the IBP_MRO_SAP9_MD_PM_S4_ERP_AddOn Task

Target IBP Master Data Type	Data Flow Name	Data Flow Description
PRODUCT	MR9_MD_PM_Product	Reads the plant-maintenance-relevant spare parts information via the /IBP/PRODUCT_TEXT extractor and transfers it to the Maintenance, Repair, and Operations planning area. To extract only those spare parts that are used in Plant Maintenance, the *Material Type* filter can be applied during the CI-DS integration. For example, the ERSA or IBAU material types can be used for filtering spare parts that are relevant for the Maintenance, Repair, and Operations planning area.
PRODUCT	MR9_MD_PM_Product_Maint	Reads the maintenance plan information via the /IBP/MRO_MAINT_ACTIVITY_TEXT extractor and transfers it to the Maintenance, Repair, and Operations planning area.
MROPARTS	MR9_MD_PM_Parts	Reads the plant-maintenance-relevant spare parts information via the /IBP/PRODUCT_TEXT extractor and transfers it to the Maintenance, Repair, and Operations planning area.
MROMAINTENANCEPLAN	MR9_MD_PM_MaintPlan_Act	Reads the maintenance plan information via the /IBP/MRO_MAINT_ACTIVITY_TEXT extractor and transfers it to the Maintenance, Repair, and Operations planning area.
CUSTOMER	MR9_MD_PM_Customer_Equi	Reads the equipment information via the /IBP/MRO_EQUIPMENT_TEXT extractor and transfers it to the Maintenance, Repair, and Operations planning area.

Target IBP Master Data Type	Data Flow Name	Data Flow Description
CUSTOMER	MR9_MD_PM_Customer_Floc	Reads the functional location information via the `/IBP/MRO_FUNCLOC_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
MROASSET	MR9_MD_PM_Asset_Floc	Reads the functional location information via the `/IBP/MRO_FUNCLOC_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
MROASSET	MR9_MD_PM_Asset_Equi	Reads the equipment information via the `/IBP/MRO_EQUIPMENT_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
LOCATION	MR9_MD_PM_Location	Reads the *Plant* and *Vendor* location types via the `/IBP/LOCATION_ATTR` extractor and transfers them to the Maintenance, Repair and Operations, planning area.
RESOURCE	MR9_MD_PM_Resource	Reads the plant maintenance resources via the `/IBP/RESOURCE_TEXT` extractor and transfers them to the Maintenance, Repair, and Operations planning area.
CUSTOMERPRODUCT	MR9_MD_PM_CustomerProd_Floc	Reads the maintenance plans data for functional locations via the `/IBP/MRO_FUNCLOC_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
CUSTOMERPRODUCT	MR9_MD_PM_CustomerProd_Equi	Reads the maintenance plans data for equipment via the `/IBP/MRO_EQUIPMENT_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.

Target IBP Master Data Type	Data Flow Name	Data Flow Description
SOURCECUSTOMER	MR9_MD_PM_CustomerSrc_Floc	Reads the plant-dependent data for maintenance plans for functional locations via the `/IBP/MRO_FUNCLOC_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
SOURCECUSTOMER	MR9_MD_PM_CustomerSrc_Equi	Reads the plant-dependent data for maintenance plans for equipment via the `/IBP/MRO_EQUIPMENT_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
LOCATIONPRODUCT	MR9_MD_PM_LocProduct	Reads the plant-maintenance-relevant spare parts – location information via the `/IBP/LOCATIONPRODUCT_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area. To extract only those spare parts – location combinations that are used in Plant Maintenance, the *Material type* filter can be applied during the CI-DS integration. For example, the `ERSA` or `IBAU` material types can be used for filtering spare parts – location combinations that are relevant for the Maintenance, Repair, and Operations planning area.
LOCATIONPRODUCT	MR9_MD_PM_LocProduct_Maint	Reads the plant-dependent data for maintenance plans via the `/IBP/MRO_MAINT_ACTIVITY_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
LOCATIONPRODUCT	MRO_MD_LocationProduct_Purch	Reads the purchasing info record data for plant-maintenance-relevant spare parts via the `/IBP/MRO_MATERIAL_VENDOR_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.

Target IBP Master Data Type	Data Flow Name	Data Flow Description
LOCATIONPARTS	MR9_MD_PM_LocParts_Purch	Reads the purchasing info record data for plant-maintenance-relevant spare parts via the `/IBP/MRO_MATERIAL_VENDOR_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
LOCATIONPARTS	MR9_MD_PM_LocParts	Reads the plant-maintenance-relevant spare parts – location information via the `/IBP/LOCATIONPRODUCT_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
SOURCELOCATION	MR9_MD_PM_SourceLoc	Reads the transportation lanes for the plant-maintenance-relevant spare parts from vendor to locations via the `/IBP/MRO_MATERIAL_VENDOR_ATTR` extractor, and transportation lanes data via the `/IBP/SOURCELOCATION_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area. To extract only those transportation lanes for the spare parts that are used in Plant Maintenance, the *Material type* filter can be applied during the CI-DS integration. For example, the `ERSA` or `IBAU` material types can be used for filtering transportation lanes for spare parts that are relevant for the Maintenance, Repair, and Operations planning area.
RESOURCELOCATION	MR9_MD_PM_ResourceLoc	Reads the location – resource information via the `/IBP/RESOURCELOCATION_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.

Target IBP Master Data Type	Data Flow Name	Data Flow Description
SOURCEPRODUCTION	MR9_MD_PM_SourceProd_Maint	Reads the source of supply data which is plant-dependent data for maintenance plans via the `/IBP/MRO_MAINT_ACTIVITY_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area. **i Note** If there is no value maintained for *Maintenance cycle short text* in the English language (or in the language that you specified as a filter), then the *Maintenance package number* value is taken.
PRODUCTIONSOURCEITEM	MR9_MD_PM_ProdSourceItem	Reads the plant-maintenance-relevant spare parts consumption data from the task list via the `/IBP/MRO_MAT_CONSUMP_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
PRODUCTIONRESOURCE	MR9_MD_PM_ProdResource	Reads the plant-maintenance-relevant resources time consumption data from the task list via the `/IBP/MRO_RES_CONSUMP_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
CURRENCY	MR9_MD_PM_Currency	Reads the currency information via the `/IBP/CURRENCY_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
EXCHANGERATE	MR9_MD_PM_ExchRate	Reads the exchange rate information via the `/IBP/EXCHANGE_RATE_ATTR` extractor and transfers it to the Maintenance, Repair and Operations planning area.
PLANNINGUNIT	MR9_MD_PM_PlanningUnit	Writes hard-coded SAP IBP MRO planning unit.

7.2.16 IBP_MRO_SAP9_PS_MD_S4_ERP_AddOn

The `IBP_MRO_SAP9_PS_MD_S4_ERP_AddOn` template contains data flows to extract MRO parts, project network activities, location, resource, project system work breakdown structure (WBS) data, location product, plant-dependent network activities data, purchasing info records, transportation lanes, resource location, capacity requirements and material components requirements for project network activities data, currency, exchange rate data from SAP S/4HANA on-premise or SAP ECC, and to load it to SAP Integrated Business Planning for Supply Chain (SAP IBP).

It uses one of the following add-ons for extraction:

- SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_s4hana_addon)
- SAP ERP, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_erp_addon)

Data Flows of the IBP_MRO_SAP9_PS_MD_S4_ERP_AddOn Task

Target IBP Master Data Type	Data Flow Name	Data Flow Description
PRODUCT	MR9_MD_PS_Product_Parts	Reads the information relevant for the materials that are used in MRO process in the Project System via the `/IBP/PRODUCT_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area. To extract only those spare parts that are used in the MRO process in the Project System, the *Material Type* filter can be applied during the CI-DS integration. For example, the `ERSA` or `IBAU` material types can be used for filtering spare parts that are relevant for the Maintenance, Repair, and Operations planning area.
MROPSNETWORKACTIVITY	MR9_MD_PS_Product_NetwAct	Reads the project network activities information via the `/IBP/MRO_PS_NETW_ACT_LOC_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
CUSTOMER	MR9_MD_PS_Customer_WBS	Reads the Project System work breakdown structure (WBS) information via the `/IBP/MRO_PS_PROJ_WBS_LOC_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.

Target IBP Master Data Type	Data Flow Name	Data Flow Description
MROPSWBSELEMENT	MR9_MD_PS_WBS	Reads the Project System work breakdown structure (WBS) information via the `/IBP/MRO_PS_PROJ_WBS_LOC_TEXT` extractor and transfers it to the Maintenance, Repair and Operations planning area.
LOCATION	MR9_MD_PS_Location	Reads the *Plant* and *Vendor* location types via the `/IBP/LOCATION_ATTR` extractor and transfers them to the Maintenance, Repair and Operations, planning area.
RESOURCE	MR9_MD_PS_Resource	Reads the plant maintenance resources via the `/IBP/RESOURCE_TEXT` extractor and transfers them to the Maintenance, Repair, and Operations planning area.
CUSTOMERPRODUCT	MR9_MD_PS_CustProd_NetwAct	Reads the project network activities data planned for the Project System work breakdown structure (WBS) via the `/IBP/MRO_PS_PROJ_WBS_LOC_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
SOURCECUSTOMER	MR9_MD_PS_CustSrc_NetwAct	Reads the plant-dependent data for project network activities planned for the Project System work breakdown structure (WBS) via the `/IBP/MRO_PS_PROJ_WBS_LOC_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.

Target IBP Master Data Type	Data Flow Name	Data Flow Description
LOCATIONPRODUCT	MR9_MD_PS_LocProduct	Reads the information relevant for the materials that are used in MRO process in the Project System at a specific location via the /IBP/ LOCATIONPRODUCT_ATTR extractor and transfers it to the Maintenance, Repair, and Operations planning area. To extract only those spare parts – location combinations that are used in the MRO process in the Project System, the *Material type* filter can be applied during the CI-DS integration. For example, the ERSA or IBAU material types can be used for filtering spare parts – location combinations that are relevant for the Maintenance, Repair, and Operations planning area.
MROPARTS	MR9_MD_PS_Parts	Reads the information relevant for the materials that are used in the MRO process in the Project System via the /IBP/PRODUCT_TEXT extractor and transfers it to the Maintenance, Repair, and Operations planning area.
LOCATIONPRODUCT	MR9_MD_PS_LocProd_NetwAct	Reads the plant-dependent network activities data via the /IBP/ MRO_PS_NETW_ACT_LOC_ATTR extractor and transfers it to the Maintenance, Repair, and Operations planning area.
LOCATIONPRODUCT	MR9_MD_PS_LocProduct_Purch	Reads the purchasing info record data for plant-maintenance-relevant spare parts via the /IBP/ MRO_MATERIAL_VENDOR_ATTR and /IBP/SOURCELOCATION_ATTR extractors and transfers it to the Maintenance, Repair, and Operations planning area.
SOURCELOCATION	MR9_MD_PS_SourceLoc	Reads the transportation lanes for the materials that are used in the MRO process in the Project System via the /IBP/SOURCELOCATION_ATTR extractor and transfers it to the Maintenance, Repair, and Operations planning area.

Target IBP Master Data Type	Data Flow Name	Data Flow Description
SOURCELOCATION	MR9_MD_PS_SourceLoc_Purch	Reads the transportation lanes for the materials that are used in the MRO process in the Project System from vendor to locations via the `/IBP/MRO_MATERIAL_VENDOR_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
RESOURCELOCATION	MR9_MD_PS_ResourceLoc	Reads the location – resource information via the `/IBP/RESOURCELOCATION_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
SOURCEPRODUCTION	MR9_MD_PS_ProdHead_NetwAct	Reads the source of supply data, which is plant-dependent network activities data via the `/IBP/MRO_PS_NETW_ACT_LOC_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
PRODUCTIONSOURCEITEM	MR9_MD_PS_ProdSourceItem	Reads the material components requirements for project network activities data via the `/IBP/MRO_PS_MAT_REQ_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
PRODUCTIONRESOURCE	MR9_MD_PS_ProdResource	Reads the network activities capacity requirements data from the task list via the `/IBP/MRO_PS_RES_REQ_ATTR` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
CURRENCY	MR9_MD_PS_Currency	Reads the currency information via the `/IBP/CURRENCY_TEXT` extractor and transfers it to the Maintenance, Repair, and Operations planning area.
EXCHANGERATE	MR9_MD_PS_ExchRate	Reads the exchange rate information via the `/IBP/EXCHANGE_RATE_ATTR` extractor and transfers it to the Maintenance, Repair and Operations planning area.

Target IBP Master Data Type	Data Flow Name	Data Flow Description
PLANNINGUNIT	MR9_MD_PS_PlanningUnit	Writes hard-coded SAP IBP MRO planning unit.

7.3 Key Figure Templates for Unified Planning Area

The templates described below are delivered by SAP. You can use them as the starting point for creating your own tasks to transfer key figure data between your SAP S/4HANA on premise or SAP ERP and SAP Integrated Business Planning.

From SAP S/4 HANA and SAP ERP, you can transfer actuals, initial inventory, open orders, planned price, and safety stock data to the unified planning area.

Make sure you load the necessary master data before you load the key figure data.

The templates filling the key figures of the unified planning area have several global variables and also a common behavior that differs from the other templates.

The common global variables are listed in the chapter *Shared Global Variables for Unified Planning Area*.

Please note the following points:

- Setting the global variable $G_BATCH_COMMAND to 'REPLACE' is forbidden by the preload script. The value 'REPLACE' is changed to 'INSERT_UPDATE' and a warning is written to the trace log. Before changing this behavior, please see Batch Command 'REPLACE' [page 228].
- All key figures, except IBP_KF_InitialInventory, are aggregated to time intervals based on technical weeks (a new technical week starts every Monday and on every first day of a month). This is the default storage level of key figure time series in the unified planning area SAPIBP1.
- The trace log contains a list of global variables and their values as well as the information if the global variables have the standard value or if they are overwritten by input parameters.
- In addition to the current load date, the last successful load date and the last successful load date with standard settings is written to the trace log.
- The selection by material types 'ROH', 'HALB', 'FERT', 'HIBE' and 'FHMI' using global variables $G_MATERIAL_TYPE_1 to $G_MATERIAL_TYPE_5 from template SOP_MD_ProductMaster is also implemented in all key figure templates described below.

7.3.1 IBP_KF_S4_ERP_AddOn

The IBP_KF_S4_ERP_AddOn template contains data flows to extract key figures from SAP S/4HANA or SAP ERP and to load it to SAP Integrated Business Planning for Supply Chain (SAP IBP). It uses one of the following add-ons for extraction:

- SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_s4hana_addon)

- SAP ERP, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_erp_addon)

For information on the prerequisites of using this template, see SAP ERP or SAP S/4HANA Add-On Integration [page 30].

The `IBP_MD_S4_ERP_AddOn` template uses the interface of the newest support package of the add-ons mentioned above. For information on adaptions that need to be done if you cannot apply the latest support package of the corresponding Add-On, see SAP ERP or SAP S/4HANA Add-On Integration [page 30].

Data Flows of the IBP_KF_S4_ERP_AddOn Task

Target Key Figure in SAP IBP	Data Flow Name	Data Flow Description
CAPASUPPLY	IBP_KF_CapaSupply_AddOn	Reads the `Capacity Supply of Resources` master data with aggregation from the `/IBP/RESCAPA_EXT` table via the `/IBP/CAPASUPPLY_KF` extractor and transfers it to the unified planning area `SAPIBP1`
INITIALINVENTORY	IBP_KF_InitialInventory_AddOn	Reads the `Initial Inventory` key figure with aggregation from the `/IBP/STOCK_KF` extractor and transfers it to the unified planning area `SAPIBP1`
MINRECEIPT	IBP_KF_MinReceipt_AddOn	Reads the `MinReceipt` key figure with aggregation from the `/IBP/ORDER_KF` extractor and transfers it to the unified planning area `SAPIBP1` **i Note** Freezing the supply plan is available for the time-series-based supply planning heuristic, the time-series-based supply planning finite heuristic, and the time-series-based supply planning optimizer. That is, you can use specific key figures to freeze the supply plan in the first few periods. You can also consider using `CONFIRMEDRECEIPT` as target key figure instead of `MINRECEIPT`. For more information, see Freezing the Supply Plan (Supply Planning Heuristic, Finite Heuristic, Optimizer).

Target Key Figure in SAP IBP	Data Flow Name	Data Flow Description
MINPRODUCTION	IBP_KF_MinProduction_AddOn	Reads the `Minimum Production` key figure with aggregation from the `/IBP/ORDER_KF` extractor and transfers it to the unified planning area SAPIBP1 **i Note** Freezing the supply plan is available for the time-series-based supply planning heuristic, the time-series-based supply planning finite heuristic, and the time-series-based supply planning optimizer. That is, you can use specific key figures to freeze the supply plan in the first few periods. You can also consider using `CONFIRMEDPRODUCTION` as target key figure instead of `MINPRODUCTION`. For more information, see Freezing the Supply Plan (Supply Planning Heuristic, Finite Heuristic, Optimizer).
EXCHANGERATE	IBP_KF_ExchangeRates_AddOn	Reads key figure data for exchange rates from the `/IBP/EXCHANGE_RATES_KF` extractor
ACTUALSQUANTITY	IBP_KF_ActualsQuantity_AddOn	Reads the `Actuals Quantity` key figure with aggregation from the `/IBP/ACTUALS_QTY_CI_KF` extractor and transfers it to the unified planning area SAPIBP1
OPENORDERSREV	IBP_KF_OpenOrders	Reads `Open Orders Rev.` key figure with aggregation from the `/IBP/OPEN_ORDERS_REV_KF` extractor and transfers it to the unified planning area SAPIBP1.

The data flows reading transactional data, that is, IBP_KF_InitialInventory_AddOn, IBP_KF_MinReceipt_AddOn, and IBP_KF_MinProduction_AddOn, have a filter condition DELTALOAD4CALLER = $G_JOB_GUID. Thanks to this, the first such data flow that is called in the task triggers the load of the latest changes from the /IBP/ORDER_DELTA and /IBP/STOCK_DELTA delta tables to the /IBP/ORDER_EXT and /IBP/STOCK_EXT extraction tables. This way, the latest changes can be read from the system without running an additional job in the add-on.

If you want to skip filtering within a job run, change the value of the $G_JOB_GUID global variable from its default setting GEN_UUID( ) to '', that is, an empty string. In case you don't want to use this feature at all, delete the filter condition in all data flows selecting from the two tables mentioned above.

How to Use This Template?

1. Import the SAP IBP staging tables for key figures of the target planning area as follows:
   1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
   2. Expand the *Timeseries Folder* in the tree.
   3. Select the planning area from which you plan to import the data, and expand it.
   4. Select the staging table with the name SOPDD_STAGING_KFTAB_<Example Planning Area><Planning Area> (do not use the table with the suffix _REP).
   5. Choose *Import*.
2. Under *Projects*, create your own project to group your data integration tasks, or select an existing project. You can, for example, reuse the project you created for the master data.
3. Create a task in your project to transfer the key figures from your SAP S/4HANA or SAP ERP system. Select IBP_KF_S4_ERP_AddOn as template.
4. As source datastore, select the datastore that represents your SAP S/4HANA or SAP ERP system.
5. As target datastore, select the datastore that represents SAP IBP, and choose *Save and Define Data Flow*.
6. You can find one target object (table) and its data flow(s) under the task you have created. Execute the following steps for each data flow.
7. If you use a planning area called other than SAPIBP1, replace the target table of the data flow as follows:
   1. Select the data flow and choose *Copy to New Target*.
   2. Enter a name for the data flow.
   3. Select the key figure staging table SOPDD_STAGING_KFTAB_<Example Planning Area><Planning Area> as target object. If you have not imported the staging table to the datastore yet, you can also perform the import in this step.
   4. Select *Copy Data Flow* or *Import and Copy Data Flow*.
8. Open the data flow for editing by choosing ▶ *Actions* ▶ *Edit* ▶.

   > **i Note**
   >
   > In the note box of the data flow, you can find instructions on adjusting the data flow to fit your needs.

9. Almost every dataflow consists of a source extractor, a SAP IBP target table, and the following intermediate steps:
   1. QueryAndFilter: Define filter conditions for the extractor and do a 1:1 mapping of the fields needed for the SAP IBP target table.
   2. MapSpecialCharsEtc: Do the mapping of the source field definition to the target field definition, including format changes, replacement of forbidden special characters, and avoiding sending fields with spaces only.
   3. Map_To_Target: Mostly 1:1 mapping of the target fields that need to be filled.
      - If required, define additional field mappings for the master data attributes in these steps. The easiest way to achieve this is to start with QueryAndFilter and drag and drop the source fields you want to map to the target structure. This automatically adds the field to the input of step MapSpecialCharsEtc and you can repeat the same there.

- You can also define additional filter conditions in step `QueryAndFilter`, but note that not every filter condition is pushed down to the source system. See Pushdown of Filter Conditions To Extractors [page 233] for more details.

10. Once you have adapted each data flow to your needs, navigate to the *Execution Properties* and set the following global variables according to your needs:
    - `$G_PLAN_AREA`: needs to be set to the name of the target planning area
    - `$G_TIMEPROFILEID_ERP`: needs to be set to the SAP IBP time profile ID used in the add-on to aggregate the key figures
    - `$G_TIMEPROFILELEVEL_ERP`: needs to be set to the SAP IBP time profile level used in the add-on to aggregate the key figures

11. Also change the corresponding row of the preload script accordingly:
    - `if $G_PLAN_AREA = '…'` (replace … with your newly set value)
    - `if $G_TIMEPROFILEID_ERP = '…'` (replace … with your newly set value)
    - `if $G_TIMEPROFILELEVEL_ERP = '…'` (replace … with your newly set value)

12. Delete the template data flows and their associated target tables from your task.

13. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

First, you need to run the task created via the `IBP_MD_S4_ERP_AddOn` template. Otherwise, the SAP IBP post processing will display error messages due to missing master data.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task and choosing *More Actions* *Promote* . Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

> i Note
>
> The data transfer between SAP S/4HANA or SAP ERP and SAP Integrated Business Planning for Supply Chain is always a complete data transfer, not a delta transfer.
>
> Transactional data are inserted and updated in SAP IBP, but not deleted.

7.3.2 IBP_KF_Actuals

The `IBP_KF_Actuals` template contains a data flow to extract actuals delivered quantity and actuals billing revenue data from SAP S/4HANA or SAP ERP and to load this data to SAP Integrated Business Planning.

Prerequisites for Creating a Task Based on the IBP_KF_Actuals Template

SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP S/4HANA system. Provide the following data for the new datastore:

   Datastore

Name and Description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP S/4HANA system	

2. Select the datastore that represents your SAP S/4HANA or SAP ERP system, and choose *Import Object by Name*. Select *Tables*, and enter the name of the tables in the *Name* field. Source tables used for this task are `LIPS`, `LIKP`, and `WB2_V_VBKR_VBRP2`.
3. Create a datastore that represents your SAP Integrated Business Planning system.
   Provide the following data for the new datastore:
   * *Type*: Integrated Business Planning
   * *Instance*: Your SAP Integrated Business Planning System

   > **i Note**
   >
   > If you cannot select your SAP Integrated Business Planning system as an instance, please contact SAP.

4. Import the staging tables that will be used for importing key figure data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Timeseries* folder in the tree, and select the planning area from which you will import the data.

3. Expand the planning area, select the staging table, and choose *Import*.

> **i Note**
>
> The name of the staging table has the pattern `SAPDD_STAGING_KFTAB_<Planning Set Name>`. If you have copied your planning area from SAP unified planning area, the planning set is a concatenation of the SAP sample name and the name of your planning area.
>
> If you use version data as a staging table, the planning set name is a concatenation of the version and the planning area name. Abbreviations may occur in the name of the table. Make sure to select tables that have this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks or select an existing project.
2. Create a task in your project to transfer the current inventory key figure data from SAP S/4HANA. Select *Use Template* and select `IBP_KF_Actuals` as template.
3. As source datastore, select the datastore that represents your SAP S/4HANA system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*. You can find the `SOPDD_STAGING_KFTAB_SAPIBP1SAPIBP1` target object (tables) and the related `IBP_ActualsRev` and `IBP_ActualsQty` data flows under the task you have created.
4. If you want to transfer data to a different planning area than the one in the template, replace the target table of the data flow. To do this, select the first data flow, for example `IBP_ActualsRev`, and choose *Copy to New Target*.
5. Enter a name for the data flow, and select the staging table of your planning area as target object. If you have not imported the staging table for your planning area to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Now a data flow copied from the task template is listed under your planning area target staging table.
7. Select the second data flow, for example `IBP_ActualsQty`, and choose ▶ *Actions* ❯ *Copy to New Target* ▶.
8. Enter a name for the second data flow, and select the staging table of your planning area as target object *Select Copy Data Flow*. Now two data flows are copied under your planning area target staging table.
9. Open the dataflow for editing by choosing ▶ *Actions* ❯ *Edit* ▶.

> **i Note**
>
> In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs.
>
> This document provides you with more detailed information about the required adjustments.

10. In the *MapToTarget* step, define the mappings for the key figures you want to transfer. In data flow `IBP_ActualsQty`, we map the actuals delivery quantity from SAP S/4 to the key figure `ACTUALSQTY` in the planning area, and in data flow `IBP_ActualsRev`, we map the actuals billing revenue from SAP S/4 to the key figure `ACTUALSREV` in the planning area..
11. Make sure you have mapped the date and at least one key figure. If you have mapped several key figures, they must refer to the same planning level. Check that you have mapped all key fields of the planning level.
12. Once you have adjusted each data flow to your needs, set the global variable `$G_PLAN_AREA` under *Execution*. Please also adapt the default value of the global variable in the preload script. You can navigate there in the edit or view mode of the task by clicking on *Execution Properties*.

If you want to transfer key figure values at an aggregated time level, specify the time profile level for upload in the `$G_TIME_PROFILE_LEVEL` variable. In this case, the key figure values will be automatically disaggregated to their base planning level. You find the numeric ID of the time profile level on the *Time Profiles* screen in the *Configuration* app.

If you don't specify a value for the `$G_TIME_PROFILE_LEVEL` parameter, key figure values are loaded at their base planning level.

The actuals quantities are aggregated to customer, product, location, and technical week level. The implementation of the aggregation to technical weeks assumes that every Monday and every first day of a month is the beginning of a new technical week. If you use a different definition of technical weeks, for example beginning the week on Sundays, you need to adapt the mapping logic. The actuals revenue has the same aggregation keys plus the currency. The reason is that in `SAPIBP1` all quantities are stored in base unit of measure, but values can be stored in several currencies. As a prerequisite the corresponding exchange rates must be loaded or replicated to the SAP IBP system.

13. Delete the template data flow and its associated target table from your task.
14. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Make sure you have imported the necessary master data before you execute the task into the sales and operations application.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Action* ▷ *Promote* ▶.. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.3.3 IBP_KF_InitialInventory

The `IBP_KF_InitialInventory` template contains a data flow to extract current inventory key figure data from SAP S/4HANA, and load it to SAP Integrated Business Planning.

Prerequisites for Creating a Task Based on the IBP_KF_InitialInventory

SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP S/4HANA system. Provide the following data for the new datastore:

Datastore

Name and Description

Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP S/4HANA system	

2. Select the datastore that represents your SAP S/4HANA system, and choose *Import Object by Name*. Select *Tables*, and enter the name of the tables in the *Name* field. Source tables used for this task are MARD and MARA.

3. Create a datastore that represents your SAP Integrated Business Planning system.
   Provide the following data for the new datastore:
   - *Type*: Integrated Business Planning
   - *Instance*: Your SAP Integrated Business Planning System

   > ℹ Note
   >
   > If you cannot select your SAP Integrated Business Planning system as an instance, please contact SAP.

4. Import the staging tables that will be used for importing key figure data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Timeseries* folder in the tree, and select the planning area from which you will import the data.
   3. Expand the planning area, select the staging table, and choose *Import*.

   > ℹ Note
   >
   > The name of the staging table has the pattern SAPDD_STAGING_KFTAB_<Planning Set Name>. If you have copied your planning area from SAP unified planning area, the planning set is a concatenation of the SAP sample name and the name of your planning area.
   >
   > If you use version data as a staging table, the planning set name is a concatenation of the version and the planning area name. Abbreviations may occur in the name of the table. Make sure to select tables that have this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks or select an existing project.
2. Create a task in your project to transfer the current inventory key figure data from SAP S/4HANA. Select *Use Template* and select IBP_KF_InitialInventory as template.
3. As source datastore, select the datastore that represents your SAP S/4HANA system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*. You can find the SOPDD_STAGING_KFTAB_SAPIBP1SAPIBP1 target object (tables) and the related IBP_InitialInventory data flow under the task you have created.

4. If you want to transfer data to a different planning area than the one in the template, replace the target table of the data flow. To do this, select the first data flow, for example `IBP_ActualsRev`, and choose *Copy to New Target*.

5. Enter a name for the data flow, and select the staging table of your planning area as target object. If you have not imported the staging table for your planning area to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.

6. Open the dataflow for editing by choosing ▶ *Actions* ❭ *Edit* ❭.

> **i Note**
>
> In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs.
>
> This document provides you with more detailed information about the required adjustments.

7. In the *MapToTarget* step, define the mappings for the key figures you want to transfer. In this task template, we map the current inventory from SAP S/4 to the key figure `INITIALINVENTORY` in the planning area.

8. Make sure you have mapped the date and at least one key figure. If you have mapped several key figures, they must refer to the same planning level. Check that you have mapped all key fields of the planning level.

9. Once you have adjusted each data flow to your needs, set the global variable `$G_PLAN_AREA` under *Execution*. Please also adapt the default value of the global variable in the preload script. You can navigate there in the edit or view mode of the task by clicking on *Execution Properties*.
If you want to transfer key figure values at an aggregated time level, specify the time profile level for upload in the `$G_TIME_PROFILE_LEVEL` variable. In this case, the key figure values will be automatically disaggregated to their base planning level. You find the numeric ID of the time profile level on the *Time Profiles* screen in the *Configuration* app.
If you don't specify a value for the `$G_TIME_PROFILE_LEVEL` parameter, key figure values are loaded at their base planning level.
The actuals quantities are aggregated to customer, product, location, and technical week level. The implementation of the aggregation to technical weeks assumes that every Monday and every first day of a month is the beginning of a new technical week. If you use a different definition of technical weeks, for example beginning the week on Sundays, you need to adapt the mapping logic. The actuals revenue has the same aggregation keys plus the currency. The reason is that in `SAPIBP1` all quantities are stored in base unit of measure, but values can be stored in several currencies. As a prerequisite the corresponding exchange rates must be loaded or replicated to the SAP IBP system.

10. Delete the template data flow and its associated target table from your task.

11. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Make sure you have imported the necessary master data before you execute the task into the sales and operations application.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Action* ❭ *Promote* ❭.. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.3.4 IBP_KF_SafetyStock

The `IBP_KF_SafetyStock` template contains a data flow to extract safety stock from SAP S/4HANA, and load it to SAP Integrated Business Planning.

Prerequisites for Creating a Task Based on the IBP_KF_SafetyStock Template

SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP S/4HANA system. Provide the following data for the new datastore:

   Datastore

Name and Description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP S/4HANA system	

2. Select the datastore that represents your SAP S/4HANA system, and choose *Import Object by Name*. Select *Tables* and enter the name of the tables in the *Name* field. Source tables used for this task are `MARC` and `MARA`.

3. Create a datastore that represents your SAP Integrated Business Planning system. Provide the following data for the new datastore:
   - *Type*: Integrated Business Planning
   - *Instance*: Your SAP Integrated Business Planning System

   > **i** Note
   >
   > If you cannot select your SAP Integrated Business Planning system as an instance, please contact SAP.

4. Import the staging tables that will be used for importing key figure data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.

2. Expand the *Timeseries* folder in the tree, and select the planning area from which you will import the data.
3. Expand the planning area, select the staging table, and choose *Import*.

> **i Note**
>
> The name of the staging table has the pattern `SAPDD_STAGING_KFTAB_<Planning Set Name>`. If you have copied your planning area from SAP unified planning area, the planning set is a concatenation of the SAP sample name and the name of your planning area.
>
> If you use version data as a staging table, the planning set name is a concatenation of the version and the planning area name. Abbreviations may occur in the name of the table. Make sure to select tables that have this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks or select an existing project.
2. Create a task in your project to transfer the current inventory key figure data from SAP S/4HANA. Select *Use Template* and select `IBP_KF_SafetyStock` as template.
3. As source datastore, select the datastore that represents your SAP S/4HANA system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*. You can find the `SOPDD_STAGING_KFTAB_SAPIBP1SAPIBP1` target object (tables) and the related `IBP_SafetySock` data flow under the task you have created.
4. If you want to transfer data to a different planning area than the one in the template, replace the target table of the data flow. To do this, select the data flow and choose *Copy to New Target*.
5. Enter a name for the data flow, and select the staging table of your planning area as target object. If you have not imported the staging table for your planning area to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Open the dataflow for editing by choosing ▶ *Actions* ❯ *Edit* ❯.

> **i Note**
>
> In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs.
>
> This document provides you with more detailed information about the required adjustments.

7. In the *MapToTarget* step, define the mappings for the key figures you want to transfer. In this task template, we map the current inventory from SAP S/4 to the key figure `SOPSAFETYSTOCK` in the planning area.
8. Make sure you have mapped the date and at least one key figure. If you have mapped several key figures, they must refer to the same planning level. Check that you have mapped all key fields of the planning level.
9. Once you have adjusted each data flow to your needs, set the global variable `$G_PLAN_AREA` under *Execution*. Please also adapt the default value of the global variable in the preload script. You can navigate there in the edit or view mode of the task by clicking on *Execution Properties*.
   If you want to transfer key figure values at an aggregated time level, specify the time profile level for upload in the `$G_TIME_PROFILE_LEVEL` variable. In this case, the key figure values will be automatically disaggregated to their base planning level. You find the numeric ID of the time profile level on the *Time Profiles* screen in the *Configuration* app.
   If you don't specify a value for the `$G_TIME_PROFILE_LEVEL` parameter, key figure values are loaded at their base planning level.
10. Delete the template data flow and its associated target table from your task.

11. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Make sure you have imported the necessary master data before you execute the task into the sales and operations application.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Action* ❯ *Promote* ◣. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.3.5 IBP_KF_OpenOrders_S4

The `IBP_KF_OpenOrders_S4` template contains a data flow to extract open order quantity and open order revenue data from SAP S/4HANA and to load it to SAP Integrated Business Planning.

This template only works for SAP S/4HANA on premise, not for older SAP ERP systems. For those systems, please use template `IBP_KF_OpenOrders_ERP` instead.

Prerequisites for Creating a Task Based on the IBP_KF_OpenOrders_S4 Template

SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP S/4HANA system. Provide the following data for the new datastore:

Datastore

Name and Description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	

Name and Description

System number and client number of your SAP S/4HANA system

2. Select the datastore that represents your SAP S/4HANA system, and choose *Import Object by Name*. Select *Tables* and enter the name of the tables in the *Name* field. Source tables used for this task are VBAP, VBAK, and MARA.

3. Create a datastore that represents your SAP Integrated Business Planning system.
   Provide the following data for the new datastore:
   - *Type*: Integrated Business Planning
   - *Instance*: Your SAP Integrated Business Planning System

> ℹ Note
>
> If you cannot select your SAP Integrated Business Planning system as an instance, please contact SAP.

4. Import the staging tables that will be used for importing key figure data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Timeseries* folder in the tree, and select the planning area from which you will import the data.
   3. Expand the planning area, select the staging table, and choose *Import*.

> ℹ Note
>
> The name of the staging table has the pattern `SAPDD_STAGING_KFTAB_<Planning Set Name>`. If you have copied your planning area from SAP unified planning area, the planning set is a concatenation of the SAP sample name and the name of your planning area.
>
> If you use version data as a staging table, the planning set name is a concatenation of the version and the planning area name. Abbreviations may occur in the name of the table. Make sure to select tables that have this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks or select an existing project.
2. Create a task in your project to transfer the current inventory key figure data from SAP S/4HANA. Select *Use Template* and select `IBP_KF_OpenOrders_S4` as template.
3. As source datastore, select the datastore that represents your SAP S/4HANA system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*. You can find the `SOPDD_STAGING_KFTAB_SAPIBP1SAPIBP1` target object (tables) and the related `IBP_OpenOrdersRev_S4` and `IBP_OpenOrdersQty_S4` data flows under the task you have created.
4. If you want to transfer data to a different planning area than the one in the template, replace the target table of the data flow. To do this, select the first data flow, for example `IBP_OpenOrdersRev_S4`, and choose *Copy to New Target*.
5. Enter a name for the data flow, and select the staging table of your planning area as target object. If you have not imported the staging table for your planning area to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Now a data flow copied from the task template is listed under your planning area target staging table.

7. Select the second data flow, for example `IBP_OpenOrdersQty_S4`, and choose ▎▶ *Actions* ❯ *Copy to New Target* ▎.

8. Enter a name for the second data flow, and select the staging table of your planning area as target object *Select Copy Data Flow*.

9. Now two data flows are copied under your planning area target staging table.

10. Open the data flow for editing by choosing ▎▶ *Action* ❯ *Edit* ▎.

> **i Note**
>
> In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs.
>
> This document provides you with more detailed information about the required adjustments.

11. In the *MapToTarget* step, define the mappings for the key figures you want to transfer. In data flow `IBP_OpenOrdersQty_S4`, we map the open order quantity from SAP S/4 to the key figure `OPENRODERSQTY` in the planning area. In data flow `IBP_OpenOrdersRev_S4`, we map the open order revenue to the key figure `OPENORDERSREV` in the planning area

12. Make sure you have mapped the date and at least one key figure. In case you have mapped several key figures, they must refer to the same planning level. Check that you have mapped all key fields of the planning level.

13. Once you have adjusted each data flow to your needs, set the global variable `$G_PLAN_AREA` under *Execution*. Please also adapt the default value of the global variable in the preload script. You can navigate there in the edit or view mode of the task by clicking on *Execution Properties*.
    If you want to transfer key figure values at an aggregated time level, specify the time profile level for upload in the `$G_TIME_PROFILE_LEVEL` variable. In this case, the key figure values will be automatically disaggregated to their base planning level. You find the numeric ID of the time profile level on the *Time Profiles* screen in the *Configuration* app.
    If you don't specify a value for the `$G_TIME_PROFILE_LEVEL` parameter, key figure values are loaded at their base planning level.

14. Delete the template data flow and its associated target table from your task.

15. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Make sure you have imported the necessary master data before you execute the task into the sales and operations application.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▎▶ *More Action* ❯ *Promote* ▎. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.3.6 IBP_KF_OpenOrders_ERP

The `IBP_KF_OpenOrders_ERP` template contains a data flow to extract open order quantity and open order revenue data from SAP ERP, and load it to SAP Integrated Business Planning.

This template only works for SAP ERP systems, not for SAP S/4HANA systems. The reason for this is as follows: The sales order item status table **VBUP** is not used any more in SAP S/4HANA. The table still exists, but it is empty. The fields from **VBUP** are copied to table **VBAP** and are filled there. So if you upgrade from SAP ERP to SAP S/4HANA either use dataflow `IBP_KF_OpenOrders_S4` or delete table **VBUP** from your copy of template `IBP_KF_OpenOrders_ERP` and use the corresponding fields from table **VBAP** instead. To do so you need to reimport table **VBAP** from the SAP S/4 HANA system to the corresponding datastore in order to get the additional fields added to the definition.

Prerequisites for Creating a Task Based on the IBP_KF_OpenOrders_S4 Template

SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP ERP system. Provide the following data for the new datastore:

   Datastore

Name and Description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP ERP system	

2. Select the datastore that represents your SAP ERP system, and choose *Import Object by Name*. Select *Tables* and enter the name of the tables in the *Name* field. Source tables used for this task are **VBAP**, **VBAK**, **VBUP**, and **MARA**.

3. Create a datastore that represents your SAP Integrated Business Planning system. Provide the following data for the new datastore:
   - *Type*: Integrated Business Planning
   - *Instance*: Your SAP Integrated Business Planning System

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, please contact SAP.

4. Import the staging tables that will be used for importing key figure data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Timeseries* folder in the tree, and select the planning area from which you will import the data.
   3. Expand the planning area, select the staging table, and choose *Import*.

> **i Note**
>
> The name of the staging table has the pattern `SAPDD_STAGING_KFTAB_<Planning Set Name>`. If you have copied your planning area from SAP unified planning area, the planning set is a concatenation of the SAP sample name and the name of your planning area.
>
> If you use version data as a staging table, the planning set name is a concatenation of the version and the planning area name. Abbreviations may occur in the name of the table. Make sure to select tables that have this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks or select an existing project.
2. 2. Create a task in your project to transfer the safety stock data from SAP ERP. Select *Use Template* and select `IBP_KF_OpenOrders_ERP` as template.
3. As source datastore, select the datastore that represents your SAP ERP system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*.
   You can find the `SOPDD_STAGING_KFTAB_SAPIBP1SAPIBP1` target object (tables) and the related `IBP_OpenOrdersRev_ERP` and `IBP_OpenOrdersQty_ERP` data flows under the task you have created.
4. If you want to transfer data to a different planning area than the one in the template, replace the target table of the data flow. To do this, select the first data flow, for example `IBP_OpenOrdersRev_ERP`, and choose *Copy to New Target*.
5. Enter a name for the data flow, and select the staging table of your planning area as target object. If you have not imported the staging table for your planning area to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Now a data flow copied from the task template is listed under your planning area target staging table.
7. Select the second data flow, for example `IBP_OpenOrdersQty_ERP`, and choose ▶ *Actions* ▶ *Copy to New Target* ▶.
8. Enter a name for the second data flow, and select the staging table of your planning area as target object *Select Copy Data Flow*.
9. Now two data flows are copied under your planning area target staging table.
10. Open the data flow for editing by choosing ▶ *Action* ▶ *Edit* ▶.

> **i Note**
>
> In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs.
>
> This document provides you with more detailed information about the required adjustments.

11. In the *MapToTarget* step, define the mappings for the key figures you want to transfer. In data flow `IBP_OpenOrdersQty_ERP`, we map the open order quantity from SAP ERP to the key figure `OPENRODERSQTY` in the planning area. In data flow `IBP_OpenOrdersRev_ERP`, we map the open order revenue to the key figure `OPENORDERSREV` in the planning area

12. Make sure you have mapped the date and at least one key figure. In case you have mapped several key figures, they must refer to the same planning level. Check that you have mapped all key fields of the planning level.

13. Once you have adjusted each data flow to your needs, set the global variable `$G_PLAN_AREA` under *Execution*. Please also adapt the default value of the global variable in the preload script. You can navigate there in the edit or view mode of the task by clicking on *Execution Properties*.
    If you want to transfer key figure values at an aggregated time level, specify the time profile level for upload in the `$G_TIME_PROFILE_LEVEL` variable. In this case, the key figure values will be automatically disaggregated to their base planning level. You find the numeric ID of the time profile level on the *Time Profiles* screen in the *Configuration* app.
    If you don't specify a value for the `$G_TIME_PROFILE_LEVEL` parameter, key figure values are loaded at their base planning level.

14. Delete the template data flow and its associated target table from your task.

15. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Make sure you have imported the necessary master data before you execute the task into the sales and operations application.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▌▶ *More Action* ❯ *Promote* ▌. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.3.7 IBP_KF_PlannedPrice

The `IBP_KF_PlannedPrice` template contains a data flow to extract the average billing document price from SAP S/4HANA and to load it to SAP Integrated Business Planning.

Prerequisites for Creating a Task Based on theIBP_KF_PlannedPrice

SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP S/4HANA system. Provide the following data for the new datastore:

Datastore

Name and Description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP S/4HANA system	

2. Select the datastore that represents your SAP S/4HANA system, and choose *Import Object by Name*. Select *Tables*, and enter the name of the tables in the *Name* field. Source tables used for this task are TCURR,MARA, WB2V, VBRK, and VBRP2.
3. Create a datastore that represents your SAP Integrated Business Planning system.
   Provide the following data for the new datastore:
   * *Type*: Integrated Business Planning
   * *Instance*: Your SAP Integrated Business Planning System

   > **i Note**
   >
   > If you cannot select your SAP Integrated Business Planning system as an instance, please contact SAP.

4. Import the staging tables that will be used for importing key figure data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Timeseries* folder in the tree, and select the planning area from which you will import the data.
   3. Expand the planning area, select the staging table, and choose *Import*.

   > **i Note**
   >
   > The name of the staging table has the pattern SAPDD_STAGING_KFTAB_<Planning Set Name>. If you have copied your planning area from SAP unified planning area, the planning set is a concatenation of the SAP sample name and the name of your planning area.
   >
   > If you use version data as a staging table, the planning set name is a concatenation of the version and the planning area name. Abbreviations may occur in the name of the table. Make sure to select tables that have this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks or select an existing project.

2. Create a task in your project to transfer the current inventory key figure data from SAP S/4HANA. Select *Use Template* and select `IBP_KF_PlannedPrice` as template.

3. As source datastore, select the datastore that represents your SAP S/4HANA system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save and Define Data Flow*. You can find the `SOPDD_STAGING_KFTAB_SAPIBP1SAPIBP1` target object (tables) and the related `IBP_PlannedPrice` data flow under the task you have created.

4. If you want to transfer data to a different planning area than the one in the template, replace the target table of the data flow. To do this, select the data flow and choose *Copy to New Target*.

5. Enter a name for the data flow, and select the staging table of your planning area as target object. If you have not imported the staging table for your planning area to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.

6. Open the dataflow for editing by choosing ▶ *Actions* ❭ *Edit* ❭.

> **i Note**
>
> In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs.
>
> This document provides you with more detailed information about the required adjustments.

7. In the *MapToTarget* step, define the mappings for the key figures you want to transfer. In this task template, we map the average price from from SAP S/4 billing documents to the key figure `PLANNEDPRICE` in the planning area.

8. Make sure you have mapped the date and at least one key figure. If you have mapped several key figures, they must refer to the same planning level. Check that you have mapped all key fields of the planning level.

9. Once you have adjusted each data flow to your needs, set the global variable `$G_PLAN_AREA` under *Execution*. Please also adapt the default value of the global variable in the preload script. You can navigate there in the edit or view mode of the task by clicking on *Execution Properties*.
If you want to transfer key figure values at an aggregated time level, specify the time profile level for upload in the `$G_TIME_PROFILE_LEVEL` variable. In this case, the key figure values will be automatically disaggregated to their base planning level. You find the numeric ID of the time profile level on the *Time Profiles* screen in the *Configuration* app.
If you don't specify a value for the `$G_TIME_PROFILE_LEVEL` parameter, key figure values are loaded at their base planning level.

10. Delete the template data flow and its associated target table from your task.

11. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Make sure you have imported the necessary master data before you execute the task into the sales and operations application.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Action* ❭ *Promote* ❭.. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.3.8 IBP_to_ERP_PIR_via_RFC

You can use the `IBP_to_ERP_PIR_via_RFC` template as a starting point to transfer consensus demand quantity from SAP Integrated Business Planning for Supply Chain to SAP S/4HANA or SAP ERP as planned independent requirements. Compared to the data flow of the `IBP_to_ERP_PIR_via_RFC_2108` template, this one is suitable for processing data in parallel, thus improving performance.

> **i Note**
>
> To use this template for integration with parallel data processing, you have to make settings in your SAP S/4HANA or SAP ERP system as well. To do so, you need SAP ERP, Supply Chain Integration Add-On for SAP Integrated Business Planning support package 19 or higher, or SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning support package 16 or higher.

You can transfer consensus demand data at product-location level, and if you maintain MRP areas in your on-premise system, you can also have them considered during integration. MRP areas can add more granularty to the basic location-product level. If you would like to use MRP areas for sorting planned independent requirements, map the MRP_AREA field in the `QueryAndFilter` step. However, if you don't want to use MRP areas, add an empty string as the value of the `MRP_AREA` field in the `Mapping` step to be sent over to the `CallRFC` step. For more information about MRP areas, see Material Requirements Planning (PP-MRP).

Time Aggregation

This template uses calendar weeks for time aggregation, as the field `PERIODID4` (level 3, calendar weeks) is mapped in the data flow. Please note that it is not sufficient to map the `PERIODID4` field in the `QueryAndFilter` transform without doing any follow-on processing with the `QueryAndFilter.PERIODID4` field. For this reason, there is the following included in the `Mapping` transform for the *ORDER_FINISH_DATE* field:

```
ifthenelse(QueryAndFilter.PERIODID4 IS NULL, QueryAndFilter.TSTFR,
QueryAndFilter.TSTFR)
```

This way, it is made sure that SAP Cloud Integration for data services requests the `PERIODID4` field from SAP IBP, which is used to calculate the `ORDER_FINISH_DATE` field.

Global Variables

Global Variable	Description
$G_AGGREGATION_UOM	It is the unit of measure to which the quantities read from SAP IBP are converted. If all your planned products have the same base unit of measure, you can just replace the default value 1BU by that unit of measure. If not, we recommend to create a unified base unit of measure in the SAP IBP system, which has a 1:1 mapping to every base unit of measure of every product. For more information, see *Defining the Planning Unit of Measure* below.
$G_LINEITEMS_PER_PACKAGE	It influences how many rows are sent together to the target system in one package. The value is not fix, as the data flow makes sure that all entries belonging to the same product location combination are sent together. In general, bigger packages reduce the overall processing time, but especially if $G_PROCESS_PIR is set, you have to make sure that the whole data package can be processed before the application times out, as the RFC call is a synchronous call and cannot be executed in the background, even if the corresponding datastore is set up that way.
$G_FIRST_SELECTION_WEEK	With these two global variables, you can influence the timeframe for which the consensus demand quantity is selected from SAP IBP and sent to your SAP ERP or SAP S/4HANA system as planned independent requirements. The first week, that is, the current week is set to zero by default, and the last week is set to 105, which is in approximately two years. These two fields are sent to the SAP IBP system by the following filter conditions: `PERIODID4 >= $G_FIRST_SELECTION_WEEK and PERIODID4 <= $G_LAST_SELECTION_WEEK` If you want to read the next 200 days in the unified planning area, you can use the filter `PERIODID0 >= 0 and PERIODID0 <= 200`. You can also combine filter conditions for different period IDs, for example, `PERIODID1 >= 0 and PERIODID0 <= 0` corresponds to year to date. However, don't use a filter `PERIODID1 = 0 and PERIODID0 <= 0` in this case because conditions with an equal are not evaluated relative to the current period, but the exact period ID is expected. If the current PERIODID0 is 1230, then `PERIODID0 = 1230` is the condition for the current period ID.

Global Variable	Description
`$G_LAST_SELECTION_WEEK`	If you want to filter on a different aggregation level, adapt the filter condition in the `QueryAndFilter` transform accordingly.
	Technically, it is also possible to filter using the `TSTFR` and `TSTTO` fields, but it can reduce performance, as the filter conditions are not pushed down to the selection of the key figure, but are only applied after data selection.

For a full list of global variables available for the `IBP_to_ERP_PIR_via_RFC` template, see Unique Global Variables for Each Template [page 188].

Prerequisites for Creating a Task Based on the IBP_to_ERP_PIR_via_RFC Template

Creating Datastores

1. Create a datastore - if it is not yet available - that represents your SAP IBP system. Provide the following data for the new datastore:

Name and Description

Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP IBP system as an instance, contact SAP.

2. Import the calculation scenario that will be used for reading consensus demand data from the planning area in SAP IBP:
   1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree, and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

      > **i Note**
      >
      > The name of the calculation scenario is of the following pattern: `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`.
      >
      > Abbreviations may occur in the name of the calculation scenario.

3. In SAP Cloud Integration for data services, create a datastore of the type File Format Group. This file is created because every data flow in SAP Cloud Integration for data services needs to have a target, and the function module cannot be a target. SAP IBP cannot be the target either, as it is already the source, and

it cannot be both a source and a target. Therefore, the file is always created, but it does not contain any useful information.

Prerequisite: The File Format Type datastore requires a root directory on the agent. You have to create the root directory, publish it on the agent, and add it to the agent's configuration before you create the following datastore.

Provide the following data for the new datastore:

Name	A name of your choice, for example, **FILES_OUT**
Description	A description of your choice, for example, **File share for output files of data flows**
Type	File Format Group
Agent	The SAP Data Services Agent in your landscape
Root Directory	Name of the directory on the agent system for storing the output files, for example, **C:\HCI_Share** for a Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. On the *File Formats* tab, choose ▶ *Create File Format* ❯ *Create from Scratch* ◼.
   Name the new file format **DUMMY_OUPUT_FILE**. If you choose a different name, you need to copy the `IBP_to_ERP_PIR` data flow to a new target later.
   Add **Dummy output file with one dummy field** as a description, and choose *OK*.
   Choose *Add Column* on the *Columns* tab. Enter **Dummy** in the *Name* field, and **Dummy field** in the **Description** field. If you choose a different name, you need to adapt the mapping in the `Target_Query` transform of the `IBP_to_ERP_PIR` data flow accordingly.
   Set the data type to VARCHAR and the length to **1**, then choose *Submit*.

5. In SAP Cloud Integration for data services, create a datastore for your SAP ERP or SAP S/4HANA system.

Name	A name of your choice, for example, **SAP ERP** or **SAP S/4HANA**
Description	A description of your choice, for example, **SAP ERP Datastore** or **SAP S/4HANA Datastore**
Type	SAP Business Suite Applications
Agent	Your data service agent
Application Server	Your SAP ERP or SAP S/4HANA application server

6. Load the definition of the /IBP/ETS_PIR_IN_PAR_RFC function module there as follows:
   1. Navigate to the *Tables* tab.
   2. Choose *Import Objects by Name*.
   3. Change the type to *Function*.
   4. In the *Name* field, enter **/IBP/ETS_PIR_IN_PAR_RFC**, and choose *OK*.

5. Check that the function module has been added to the list of tables.

> **i Note**
>
> If the data flow has several configurations, make sure the configuration of the system for loading the function module is the default one. If it isn't, make it the default and don't forget to save the changes.

7. Load the definition of the `/IBP/ETS_PIR_PROCESS_BATCH` function module as well.

Defining the Planning Unit of Measure

To be able to run the `IBP_to_ERP_PIR_via_RFC`, a planning unit of measure (UoM) is required in SAP IBP. The conversion factor 1:1 needs to be maintained between this planning unit of measure and the base unit of measure for all products in SAP IBP. By default, this planning UoM is called `1BU` in the SAP Cloud Integration for data services template.

To make sure that your planning UoM is adapted to the template, follow the instructions that apply to your case:

- If there is a planning UoM already available in SAP IBP called **1BU**, no further steps are required.
- If there is a planning UoM already available in SAP IBP, but it is called differently, change the value of the `$G_AGGREGATION_UOM` global variable in your SAP Cloud Integration for data services task to the name of your own planning UoM.
- If there is no planning UoM available in SAP IBP yet, create one in one of the following ways:
  - Create it manually and maintain the conversion factor between this planning UoM and the base UoM as 1:1 in SAP IBP.
  - Create it using the `IBP_MD_UOM_w_Text_AddOn` and `IBP_MD_UOMConversionFactor_AddOn` data flows of the `IBP_MD_S4_ERP_AddOn` template and create the conversion factor 1:1 between this planning UoM and the base UoM using the SAP Cloud Integration for data services task.
    If the `$G_UNIFIED_BASE_UOM` global parameter is not set to an empty string, the data flows add both a unified base unit of measure to SAP IBP with the name `$G_UNIFIED_BASE_UOM` and the necessary 1:1 mappings.

If there is a unit of measure into which every product can be converted, you can also set the value of the `$G_AGGREGATION_UOM` global variable to that unit of measure and map the `CT_PIR_IN-UNIT_OF_MEASURE` field to the same value. By default, this field is mapped to `' '` in the `Mapping` step, and the base unit of measure is used in the target system.

How to Use This Template?

1. Under *Projects*, open your project, or create a new one.
2. Create a task in your project to transfer the consensus demand quantity data to SAP ERP or SAP S/4HANA. Select *Use Template*, and select `IBP_to_ERP_PIR_via_RFC` as template.
3. As source datastore, select the datastore that represents your SAP IBP system. As target datastore, select the datastore of the type File Format Group that you have previously created. Choose *Save and Define Data Flow*.
   In case you have named the file format other than **DUMMY_OUTPUT_FILE**, select the data flow, and choose *Copy to new target*. Enter a name for the data flow, select the target file format, and choose *Copy Data Flow*. Then, delete the original data flow and the target table `DUMMY_OUTPUT_FILE`.

4. Open the data flow for editing by choosing ▶ *Actions* ▶ *Edit* ▶.
5. Follow the instructions in the note boxes of the data flow to create mappings and adjust the data flow to fit your needs.
6. Under *Execution Properties*, you can modify the default values of the following global variables:
   - `$G_LINEITEMS_PER_PACKAGE`
   - `$G_FIRST_SELECTION_WEEK`
   - `$G_LAST_SELECTION_WEEK`
   - `$G_VERSION_ACTIVE`
   - `$G_VERSION_NUMBER`
   - `$G_REQ_TYPE`
   - `$G_REQ_PLAN_NUMBER`
   - `$G_AGGREGATION_UOM`
   - `$G_PROCESS_PIR`
   - `$G_DATE_TYPE`
   - `$G_DELETE_OLD`

   If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:
   ```
   IF (<global variable> = '<default value>')
   ```
7. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ▶ *Promote* ▶. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

Error Handling

The `/IBP/ETS_PIR_IN_PAR_RFC` and `/IBP/ETS_PIR_PROCESS_BATCH` function modules also have exporting tables with messages generated while creating or processing the staging table entries. These messages can have different types (abort, error, warning, information, and success). The `CollectMessages` transform makes sure the messages are sorted by type, showing the most severe first. The `PrintMessages` transform prints the messages to the trace log, while the `AggregateMsgTypes` transform makes sure that all messages are written before an exception is thrown. `StatusFieldsOnly` collects the results of the `EV_STATUS` and `AL_RFC_RETCODE` status fields. In the `ExceptionIfNeeded` transform, an exception is thrown in case `AL_RFC_RETCODE` is not equal `RFC_OK`, `EV_STATUS` is greater than zero, or `ET_MESSAGES` contains abort messages. You can change this behavior by modifying the field mapping of the `EV_STATUS` field in the `ExceptionIfNeeded` transform. For example, you can achieve that the status of the task is set to error if `ET_MESSAGES` contains error messages by removing the comments in the commented rows of the mapping.

If an error occurs during the RFC call, it does not necessarily lead to an error status. Therefore, processing of the output data of the `/IBP/ETS_PIR_IN_PAR_RFC` and `/IBP/ETS_PIR_PROCESS_BATCH` function modules is needed. The exporting parameter `AL_RFC_RETCODE` of the `CallRFC` and `CallRFCProcess` transforms is a parameter added by the SAP Cloud Integration for data services framework. If it does not contain the `RFC_OK` value, there has been a technical error while calling the function module. The function modules themselves have the exporting parameter `EV_STATUS`, which has a value greater than zero if a technical error occurred during package processing in SAP S/4HANA or SAP ERP within the function module call.

> **i Note**
>
> Note that the parameter `AL_RFC_RETCODE` is not part of the function module interface, but is added to indicate if the function call was technically successful (`RFC_OK`) or not (other value).

7.3.9 IBP_to_ERP_PIR_via_RFC_2108

You can use the `IBP_to_ERP_PIR_via_RFC_2108` template as a starting point to transfer consensus demand quantity from SAP Integrated Business Planning for Supply Chain (SAP IBP) to SAP S/4HANA on premise or SAP ERP as planned independent requirements.

> **i Note**
>
> If you are using SAP ERP, Supply Chain Integration Add-On for SAP Integrated Business Planning support package 18 or lower, or SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning support package 15 or lower, use this template for integrating planned independent requirements. For higher SP versions, we recommend to use the template `IBP_to_ERP_PIR_via_RFC` instead, which provides additional features.

You can transfer consensus demand data at product-location level, and if you maintain MRP areas in your on-premise system, you can also have them considered during integration. MRP areas can add more granularty to the basic location-product level. If you would like to use MRP areas for sorting planned independent requirements, map the `MRP_AREA` field in the `QueryAndFilter` step. However, if you don't want to use MRP areas, add an empty string as the value of the `MRP_AREA` field in the `PrepareRFC` step to be sent over to the `CallRFC` step. For more information about MRP areas, see Material Requirements Planning (PP-MRP).

Time Aggregation

This template uses calendar weeks for time aggregation, as the field `PERIODID4` (level 3, calendar weeks) is mapped in the data flow. Please note that it is not sufficient to map the `PERIODID4` field in the `QueryAndFilter` transform without doing any follow-on processing with the `QueryAndFilter.PERIODID4` field. For this reason, there is the following included in the `Mapping` transform for the *ORDER_FINISH_DATE* field:

```
ifthenelse(QueryAndFilter.PERIODID4 IS NULL, QueryAndFilter.TSTFR,
QueryAndFilter.TSTFR)
```

This way, it is made sure that SAP Cloud Integration for data services requests the `PERIODID4` field from SAP IBP, which is used to calculate the `ORDER_FINISH_DATE` field.

Global Variables

Global Variable	Description
`$G_AGGREGATION_UOM`	It is the unit of measure to which the quantities read from SAP IBP are converted. If all your planned products have the same base unit of measure, you can just replace the default value `1BU` by that unit of measure. If not, we recommend to create a unified base unit of measure in the SAP IBP system, which has a 1:1 mapping to every base unit of measure of every product. For more information, see *Defining the Planning Unit of Measure* below.
`$G_LINEITEMS_PER_PACKAGE`	It influences how many rows are sent together to the target system in one package. The value is not fix, as the data flow makes sure that all entries belonging to the same product location combination are sent together. In general, bigger packages reduce the overall processing time, but especially if `$G_PROCESS_PIR` is set, you have to make sure that the whole data package can be processed before the application times out, as the RFC call is a synchronous call and cannot be executed in the background, even if the corresponding datastore is set up that way.
`$G_FIRST_SELECTION_WEEK`	With these two global variables, you can influence the timeframe for which the consensus demand quantity is selected from SAP IBP and sent to your SAP ERP or SAP S/4HANA system as planned independent requirements. The first week, that is, the current week is set to zero by default, and the last week is set to 105, which is in approximately two years. These two fields are sent to the SAP IBP system by the following filter conditions: `PERIODID4 >= $G_FIRST_SELECTION_WEEK and PERIODID4 <= $G_LAST_SELECTION_WEEK` If you want to read the next 200 days in the unified planning area, you can use the filter `PERIODID0 >= 0 and PERIODID0 <= 200`. You can also combine filter conditions for different period IDs, for example, `PERIODID1 >= 0 and PERIODID0 <= 0` corresponds to year to date. However, don't use a filter `PERIODID1 = 0 and PERIODID0 <= 0` in this case because conditions with an equal are not evaluated relative to the current period, but

Global Variable	Description
`$G_LAST_SELECTION_WEEK`	the exact period ID is expected. If the current PERIODID0 is 1230, then `PERIODID0 = 1230` is the condition for the current period ID.
	If you want to filter on a different aggregation level, adapt the filter condition in the `QueryAndFilter` transform accordingly.
	Technically, it is also possible to filter using the `TSTFR` and `TSTTO` fields, but it can reduce performance, as the filter conditions are not pushed down to the selection of the key figure, but are only applied after data selection.

For a full list of global variables available for the `IBP_to_ERP_PIR_via_RFC_2108` template, see Unique Global Variables for Each Template [page 188].

Prerequisites for Creating a Task Based on the IBP_to_ERP_PIR_via_RFC_2108 Template

Creating Datastores

1. Create a datastore - if it is not yet available - that represents your SAP IBP system. Provide the following data for the new datastore:

Name and Description	
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP IBP system as an instance, contact SAP.

2. Import the calculation scenario that will be used for reading consensus demand data from the planning area in SAP IBP:
   1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree, and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

      > **i Note**
      >
      > The name of the calculation scenario is of the following pattern: `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`.
      >
      > Abbreviations may occur in the name of the calculation scenario.

3. In SAP Cloud Integration for data services, create a datastore of the type File Format Group. This file is created because every data flow in SAP Cloud Integration for data services needs to have a target, and the function module cannot be a target. SAP IBP cannot be the target either, as it is already the source, and it cannot be both a source and a target. Therefore, the file is always created, but it does not contain any useful information.

   Prerequisite: The File Format Type datastore requires a root directory on the agent. You have to create the root directory, publish it on the agent, and add it to the agent's configuration before you create the following datastore.

   Provide the following data for the new datastore:

Name	A name of your choice, for example, **FILES_OUT**
Description	A description of your choice, for example, **File share for output files of data flows**
Type	File Format Group
Agent	The SAP Data Services Agent in your landscape
Root Directory	Name of the directory on the agent system for storing the output files, for example, **C:\HCI_Share** for a Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. On the *File Formats* tab, choose ▶ *Create File Format* ▶ *Create from Scratch* ▶.
   Name the new file format **DUMMY_OUPUT_FILE**. If you choose a different name, you need to copy the IBP_to_ERP_PIR data flow to a new target later.
   Add **Dummy output file with one dummy field** as a description, and choose *OK*.
   Choose *Add Column* on the *Columns* tab. Enter **Dummy** in the *Name* field, and **Dummy field** in the **Description** field. If you choose a different name, you need to adapt the mapping in the Target_Query transform of the IBP_to_ERP_PIR data flow accordingly.
   Set the data type to VARCHAR and the length to **1**, then choose *Submit*.

5. In SAP Cloud Integration for data services, create a datastore for your SAP ERP or SAP S/4HANA system.

Name	A name of your choice, for example, **SAP ERP** or **SAP S/4HANA**
Description	A description of your choice, for example, **SAP ERP Datastore** or **SAP S/4HANA Datastore**
Type	SAP Business Suite Applications
Agent	Your data service agent
Application Server	Your SAP ERP or SAP S/4HANA application server

Load the definition of the /IBP/ETS_PIR_IN_RFC function module there as follows:

1. Navigate to the *Tables* tab.

2. Choose *Import Objects by Name*.
3. Change the type to *Function*.
4. In the *Name* field, enter **/IBP/ETS_PIR_IN_RFC**, and choose *OK*.
5. Check that the function module has been added to the list of tables.

> **i** Note
>
> If the data flow has several configurations, make sure the configuration of the system for loading the function module is the default one. If it isn't, make it the default and don't forget to save the changes.

Defining the Planning Unit of Measure

To be able to run the `IBP_to_ERP_PIR_via_RFC`, a planning unit of measure (UoM) is required in SAP IBP. The conversion factor 1:1 needs to be maintained between this planning unit of measure and the base unit of measure for all products in SAP IBP. By default, this planning UoM is called `1BU` in the SAP Cloud Integration for data services template.

To make sure that your planning UoM is adapted to the template, follow the instructions that apply to your case:

- If there is a planning UoM already available in SAP IBP called **1BU**, no further steps are required.
- If there is a planning UoM already available in SAP IBP, but it is called differently, change the value of the `$G_AGGREGATION_UOM` global variable in your SAP Cloud Integration for data services task to the name of your own planning UoM.
- If there is no planning UoM available in SAP IBP yet, create one in one of the following ways:
  - Create it manually and maintain the conversion factor between this planning UoM and the base UoM as 1:1 in SAP IBP.
  - Create it using the `IBP_MD_UOM_w_Text_AddOn` and `IBP_MD_UOMConversionFactor_AddOn` data flows of the `IBP_MD_S4_ERP_AddOn` template and create the conversion factor 1:1 between this planning UoM and the base UoM using the SAP Cloud Integration for data services task.
    If the `$G_UNIFIED_BASE_UOM` global parameter is not set to an empty string, the data flows add both a unified base unit of measure to SAP IBP with the name `$G_UNIFIED_BASE_UOM` and the necessary 1:1 mappings.

If there is a unit of measure into which every product can be converted, you can also set the value of the `$G_AGGREGATION_UOM` global variable to that unit of measure and map the `CT_PIR_IN-UNIT_OF_MEASURE` field to the same value. By default, this field is mapped to `' '` in the `Mapping` step, and the base unit of measure is used in the target system.

How to Use This Template?

1. Under *Projects*, open your project, or create a new one.
2. Create a task in your project to transfer the consensus demand quantity data to SAP ERP or SAP S/4HANA. Select *Use Template*, and select `IBP_to_ERP_PIR_via_RFC_2108` as template.
3. As source datastore, select the datastore that represents your SAP IBP system. As target datastore, select the datastore of the type File Format Group that you have previously created. Choose *Save and Define Data Flow*.

In case you have named the file format other than **DUMMY_OUTPUT_FILE**, select the data flow, and choose *Copy to new target*. Enter a name for the data flow, select the target file format, and choose *Copy Data Flow*. Then, delete the original data flow and the target table DUMMY_OUTPUT_FILE.

4. Open the data flow for editing by choosing ▌ *Actions* ❯ *Edit* ▌.

> **i Note**
>
> In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

5. Replace the source table with your own calculation view and recreate the mapping arrow from the source table to the QueryAndFilter transform.
6. Delete the CallRFC transform in the IBP_to_ERP_PIR_via_RFC_2108 data flow, replace it with a transform of type *Web Service or Function Call*, and name it **CallRFC** again.
7. Create the mapping arrows from PrepareRFC to **CallRFC** and from **CallRFC** to GetRFCOutput.
8. Choose the **CallRFC** transform. Choose *Select Function: Web Service or RFC Function* on the output node CallRFC. Select the function module you have uploaded from your SAP S/4HANA or SAP ERP datastore, and map all first-level nodes of PrepareRFC (for example, CT_PIR_IN and IV_PROCESS_PIR) to the corresponding nodes of CallRFC.
9. Under *Execution Properties*, you can modify the default values of the following global variables:
   - $G_LINEITEMS_PER_PACKAGE
   - $G_FIRST_SELECTION_WEEK
   - $G_LAST_SELECTION_WEEK
   - $G_VERSION_ACTIVE
   - $G_VERSION_NUMBER
   - $G_REQ_TYPE
   - $G_REQ_PLAN_NUMBER
   - $G_AGGREGATION_UOM
   - $G_PROCESS_PIR
   - $G_DATE_TYPE
   - $G_DELETE_OLD

   If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:
   ```
   IF (<global variable> = '<default value>')
   ```
10. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▌ *More Actions* ❯ *Promote* ▌. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

Error Handling

The `/IBP/ETS_PIR_IN_RFC` function module also has an exporting table with messages generated while creating or processing the staging table entries. These messages can have different types (abort, error, warning, information, and success). The `CollectMessages` transform makes sure the messages are sorted by type, showing the most severe first. The `PrintMessages` transform prints the messages to the trace log, while the `AggregateMsgTypes` transform makes sure that all messages are written before an exception is thrown. `StatusFieldsOnly` collects the results of the `EV_STATUS` and `AL_RFC_RETCODE` status fields. In the `ExceptionIfNeeded` transform, an exception is thrown in case `AL_RFC_RETCODE` is not equal `RFC_OK`, `EV_STATUS` is greater than zero, or `ET_MESSAGES` contains abort messages. You can change this behavior by modifying the field mapping of the `EV_STATUS` field in the `ExceptionIfNeeded` transform. For example, you can achieve that the status of the task is set to error if `ET_MESSAGES` contains error messages by removing the comments in the commented rows of the mapping.

If an error occurs during the RFC call, it does not necessarily lead to an error status. Therefore, processing of the output data of the `/IBP/ETS_PIR_IN_RFC` function module is needed. Exporting parameter `AL_RFC_RETCODE` of the `CallRFC` transform is a parameter added by the SAP Cloud Integration for data services framework. If it does not contain the `RFC_OK` value, there has been a technical error while calling the function module. The function module itself has an exporting parameter `EV_STATUS`, which has a value greater than zero if a technical error occurred during package processing in SAP S/4HANA or SAP ERP within the function module call.

> ℹ **Note**
>
> Note that the parameter `AL_RFC_RETCODE` is not part of the function module interface, but is added to indicate if the function call was technically successful (`RFC_OK`) or not (other value).

7.4 Key Figure Templates for Application-Specific Planning Areas

SAP delivers templates you can use as the starting point for creating your own tasks to transfer key figure data between your SAP S/4 HANA, SAP ERP, SAP APO, SAP SCM, or SAP Demand Signal Management system and SAP Integrated Business Planning.

You can transfer data between your SAP S/4 HANA, SAP ERP, SAP APO, SAP SCM, or SAP Demand Signal Management system and the sales and operation or the demand application in both directions:

- From SAP S/4 HANA and SAP ERP, you can transfer actuals quantity, inventory, and open orders data to the sales and operations application.
- From Demand Planning in SAP APO, you can transfer demand planning key figures to SAP Integrated Business Planning for demand or to SAP Integrated Business Planning for sales and operations.
- From SAP Integrated Business Planning for demand, you can transfer sensed demand data to SAP APO's Supply Network Planning.
- From SAP Demand Signal Management, you can transfer POS data to SAP Integrated Business Planning for demand.

Make sure you load the necessary master data before you load the key figure data.

7.4.1 IBP_DDR_KF_ERP_AddOn

The `IBP_DDR_KF_ERP_AddOn` template contains data flows to extract key figures from SAP S/4HANA on-premise or SAP ECC and to load it to a demand-driven replenishment planning area of SAP Integrated Business Planning copied from the `SAP8` example planning area. It uses one of the following add-ons for extraction:

- SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_s4hana_addon)
- SAP ERP, supply chain integration add-on for SAP Integrated Business Planning (for more information, see http://help.sap.com/ibp_erp_addon)

For information on the prerequisites of using this template, see SAP ERP or SAP S/4HANA Add-On Integration [page 30].

The `IBP_DDR_KF_ERP_AddOn` template uses the interface of the newest support package of the add-ons mentioned above. For information on adaptions that need to be done if you cannot apply the latest support package of the corresponding Add-On, see SAP ERP or SAP S/4HANA Add-On Integration [page 30].

Data Flows of the IBP_DDR_KF_ERP_AddOn Task

Target Key Figure in IBP	Data Flow Name	Data Flow Description
DDONHANDINV	IBP_DDR_KF_OnHandInventory	Reads the `On Hand Inventory` key figure with aggregation from the `/IBP/STOCK_KF` extractor and transfer it to the demand-driven replenishment planning area `SAP8` in IBP
DDTOTALDEMANDQTYFUTURE	IBP_DDR_KF_TotalDemand	Reads the `Total Demand` key figure with aggregation from the `/IBP/TOTAL_DEMAND_QTY_CI_KF` extractor and transfers it to the demand-driven replenishment planning area `SAP8` in IBP
DDTOTALORDERQTYHISTORY	IBP_DDR_KF_ActualsQuantity	Reads the `Actuals Quantity` key figure with aggregation from the `/IBP/ACTUALS_QTY_CI_KF` extractor and transfer it to the demand-driven replenishment planning area `SAP8` in IBP

Target Key Figure in IBP	Data Flow Name	Data Flow Description
DDSUPORDERNONCONFQTY	IBP_DDR_KF_SupplyOrderNonConf	Reads the `Non-Confirmed Supply Orders` key figure with aggregation from the `/IBP/ORDER_KF` extractor and transfers it to the demand-driven replenishment planning area `SAP8` in IBP
DDSUPORDERCONFQTY	IBP_DDR_KF_SupplyOrderConfirmed	Reads the `Confirmed Supply Order` key figure with aggregation from the `/IBP/ORDER_KF` extractor and transfers it to the demand-driven replenishment planning area `SAP8` in IBP
DDDEMORDERCONFQTY	IBP_DDR_KF_ConfirmedOrder	Reads the `Confirmed Orders` key figure with aggregation from the `/IBP/ORDER_KF` extractor and transfers it to the demand-driven replenishment planning area `SAP8` in IBP

> **i Note**
>
> Previously, demand orders and forecasts were selected in two separate data flows and written to two different key figures. Now, the `Total Demand` key figure contains the sum of both and is filled using the new extractor `/IBP/TOTAL_DEMAND_QTY_CI_KF`.

How to Use This Template?

1. Import the SAP Integrated Business Planning staging tables for key figures of the target planning area as follows:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Timeseries Folder* in the tree.
   3. Select the planning area from which you plan to import the data, and expand it.
   4. Select the staging table with the name `SOPDD_STAGING_KFTAB_<Example Planning Area><Planning Area>` (do not use the table with the suffix `_REP`).
   5. Choose *Import*.
2. Under *Projects*, create your own project to group your data integration tasks, or select an existing project. You can, for example, reuse the project you created for the master data.
3. Create a task in your project to transfer the key figures from your SAP S/4HANA or SAP ECC system. Select `IBP_DDR_KF_ERP_AddOn` as template.
4. As source datastore, select the datastore that represents your SAP S/4HANA or SAP ECC system.
5. As target datastore, select the datastore that represents SAP Integrated Business Planning, and choose *Save and Define Data Flow*.

6. You can find one target object (table) and its data flow(s) under the task you have created. Execute the following steps for each data flow.

7. If you use a planning area called other than `SAPIBP1`, replace the target table of the data flow as follows:

   1. Select the data flow, and choose *Copy to New Target*.

   2. Enter a name for the data flow.

   3. Select the key figure staging table `SOPDD_STAGING_KFTAB_<Example Planning Area><Planning Area>` as target object. If you have not imported the staging table to the datastore yet, you can also perform the import in this step.

   4. Select *Copy Data Flow* or *Import and Copy Data Flow*.

8. Open the data flow for editing by choosing ▌▶ *Actions* ❯ *Edit* ▌.

> **i Note**
>
> In the note box of the data flow, you can find instructions on adjusting the data flow to fit your needs.

9. Almost every dataflow consists of a source extractor, an IBP target table, and the following intermediate steps:

   1. `QueryAndFilter`: Define filter conditions for the extractor and do a 1:1 mapping of the fields needed for the IBP target table.

   2. `MapSpecialCharsEtc`: Do the mapping of the source field definition to the target field definition, including format changes, replacement of forbidden special characters, and avoiding sending fields with spaces only.

   3. `Map_To_Target`: Mostly 1:1 mapping of the target fields that need to be filled.

      - If required, define additional field mappings for the master data attributes in these steps. The easiest way to achieve this is to start with `QueryAndFilter` and drag and drop the source fields you want to map to the target structure. This automatically adds the field to the input of step `MapSpecialCharsEtc` and you can repeat the same there.

      - You can also define additional filter conditions in step `QueryAndFilter`, but note that not every filter condition is pushed down to the source system. See Pushdown of Filter Conditions To Extractors [page 233] for more details.

10. Once you have set each data flow, navigate to *Execution Properties* and set the following global variables according to your needs:

    - `$G_PLAN_AREA` (set it to the name of the target planning area)

    - `$G_MRPTYPE_DDR` (set it to the ERP MRP type, which is relevant for Demand-Driven Replenishment)

11. Change the corresponding rows of the preload script accordingly:

    - If `$G_PLAN_AREA = '…'` (replace … with your newly set value)

    - If `$G_MRPTYPE_DDR = '…'` (replace … with your newly set value)

12. Delete the template data flows and their associated target tables from your task.

13. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

First, you need to run the task created via the `IBP_DDR_MD_ERP_AddOn` template. Otherwise, the IBP post processing will display error messages due to missing master data.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task and choosing ▶ *More Actions* ❯ *Promote* ◀. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

> ℹ Note
>
> The data transfer between SAP S/4HANA / SAP ECC and SAP Integrated Business Planning is always a complete data transfer, not a delta transfer.
>
> Master data is only inserted and updated in IBP, but not deleted.

7.4.2 IBP_DDR_to_ERP_AddOn_via_RFC

You can use the `IBP_DDR_to_ERP_AddOn_via_RFC` template as a starting point to transfer the input for the MRP run in ERP from the demand-driven replenishment planning area in SAP Integrated Business Planning to the IBP add-on in SAP ECC.

> ℹ Note
>
> Please note that this template replaces the previous `IBP_DDR_to_ERP_AddOn_via_WS` and `IBP_DDR_KF_to_ERP_AddOn_via_WS` templates that contained web-service-based data flows. We strongly recommend to replace your existing data flows that are based on these templates by ones based on the `IBP_DDR_to_ERP_AddOn_via_RFC` template, as the new template is easier to maintain and more stable.

The template consists of the following data flows:

Data Flows of IBP_DDR_to_ERP_AddOn_via_RFC

Data Flow Name	Description	Source Key Figure and Master Data Fields	Target Function Module and Table
IBP_DDR_to_ERP_DDPrdLoc	Must run first. Reads demand-driven product locations from IBP and transfers them to SAP ERP supply chain integration add-on for SAP Integrated Business Planning using the /IBP/EDD_PL_FILL RFC-enabled function module.	Key figure fields: • PRDID • LOCID • RECDECOUPPOINTIND • DDORDERREC • DECOUPLEDLEADTIMEDAYS • DDBUFFERSTATUS • DDNETFLOWPOSITION • REDZONEBASE • TOPOFRED • TOPOFYELLOW • TOPOFGREEN • PERIODID0 • TSTFR	/IBP/EDD_PL_FILL /IBP/EDD_PL
IBP_DDR_to_ERP_DsDcplngPntDelete	Reads location products without downstream decoupling points from the IBP demand-driven replenishment planning area and transfers them to SAP ERP supply chain integration add-on for SAP Integrated Business Planning using the /IBP/EDD_DSDCP_FILL RFC-enabled function module (for deleting remaining entries)	Master data fields: • LOCID • PRDID	/IBP/EDD_DS_DCP_FILL /IBP/EDD_DS_DCP
IBP_DDR_to_ERP_DsDcplngPnt	Reads downstream decoupling points from the IBP demand-driven replenishment planning area and transfers them to SAP ERP supply chain integration add-on for SAP Integrated Business Planning using the /IBP/EDD_DSDCP_FILL RFC-enabled function module	Master data fields: • LOCID • LOCIDMAPDCPPT • PRDID • PRDIDMAPDCPPT	/IBP/EDD_DS_DCP_FILL /IBP/EDD_DS_DCP

Data Flow Name	Description	Source Key Figure and Master Data Fields	Target Function Module and Table
IBP_DDR_to_ERP_BufferLvl	Reads time-dependent buffer levels from the IBP demand-driven replenishment planning area and transfers them to SAP ERP supply chain integration add-on for SAP Integrated Business Planning using the `/IBP/EDD_BLT_FILL` RFC-enabled function module	Master data fields: • PRDID • LOCID Key figure fields: • TSTFR • GROUP_COUNTER • DDEXECTOPOFGREEN • DDEXECTOPOFLOWYELLOW • DDEXECTOPOFLOWRED • PERIODID0	/IBP/EDD_BLT_FILL /IBP/EDD_BLT
IBP_DDR_to_ERP_OrderCorrDelete	Reads location products without order corrections from the IBP demand-driven replenishment planning area and transfers them to SAP ERP supply chain integration add-on for SAP Integrated Business Planning using the `/IBP/EDD_COR_FILL` RFC-enabled function module	Master data fields: • PRDID • LOCID Key figure fields: • TSTFR • PERIODID0 • DDOPENSUPPLY • DDORDERSPIKE • DECOUPLEDLEADTIMEDAYS	/IBP/EDD_COR_FILL /IBP/EDD_COR
IBP_DDR_to_ERP_OrderCorrections	Reads time-dependent order corrections from the IBP demand-driven replenishment planning area and transfers them to SAP ERP supply chain integration add-on for SAP Integrated Business Planning using the `/IBP/EDD_COR_FILL` RFC-enabled function module	Master data fields: • PRDID • LOCID Key figure fields: • TSTFR • PERIODID0 • DDOPENSUPPLY • DDORDERSPIKE • DECOUPLEDLEADTIMEDAYS	/IBP/EDD_COR_FILL /IBP/EDD_COR

If you decide not to copy this template for creating your task, or if you need to copy the data flows to other target files, make sure that the target file of the `IBP_DDR_to_ERP_DDPrdLoc` data flow is the first in the execution order. To set the order, do the following:

1. Select the target table.
2. Choose ▶ *Actions* ▶ *Manage target order* ▶.

3. Move the target file of the `IBP_DDR_to_ERP_DDPrdLoc` data flow to the first position. The order of the other objects doesn't matter.

Prerequisites for Creating a Task Based on the `IBP_DDR_to_ERP_AddOn_via_RFC` **Template**

Creating Datastores

1. If not yet available, create a datastore that represents your SAP Integrated Business Planning system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP Integrated Business Planning system

> **i** Note
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading demand-driven replenishment data from the planning area in SAP Integrated Business Planning:

   1. Select the datastore that represents your SAP Integrated Business Planning system and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

      > **i** Note
      >
      > The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`. Abbreviations may occur in the name of the calculation scenario.

   4. Also import the necessary master data tables `SOPMD_<prefix>LOCATIONPRODUCT` and `SOPMD_<prefix>DECOUPLINGPOINTSMAP`, where **`<prefix>`** needs to be replaced by the master data prefix used in the relevant planning area.

3. In SAP Cloud Integration for data services, create a datastore of the type File Format Group. This file is created because every data flow in SAP Cloud Integration for data services needs to have a target, and the function module cannot be a target. IBP cannot be the target either, as it is already the source, and it cannot be both a source and a target. Therefore, the file is always created, but it does not contain any useful information.
   Prerequisite: The File Format Type datastore requires a root directory on the agent. You have to create the root directory, publish it on the agent, and add it to the agent's configuration before you create the following datastore.

Provide the following data for the new datastore:

Name	A name of your choice, for example, **FILES_OUT**
Description	A description of your choice, for example, **File share for output files of data flows**
Type	File Format Group
Agent	The SAP Data Services Agent in your landscape
Root Directory	Name of the directory on the agent system for storing the output files, for example, **C:\HCI_Share** for a Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. On the *File Formats* tab, choose ▶ *Create File Format* ▶ *Create from Scratch* ▶.
   Name the new file format **DUMMY_OUPUT_FILE**. If you choose a different name, you need to copy the data flows to a new target later.
   Add **Dummy output file with one dummy field** as a description, and choose *OK*.
   Choose *Add Column* on the *Columns* tab. Enter **Dummy** in the *Name* field, and **Dummy field** in the **Description** field. If you choose a different name, you need to adapt the mapping in the Target_Query transform of the data flows accordingly.
   Set the data type to VARCHAR and the length to **1**, then choose *Submit*.
5. In SAP Cloud Integration for data services, create a datastore for your SAP ERP or SAP S/4HANA system.

Name	A name of your choice, for example, **SAP ERP** or **SAP S/4HANA**
Description	A description of your choice, for example, **SAP ERP Datastore** or **SAP S/4HANA Datastore**
Type	SAP Business Suite Applications
Agent	Your data service agent
Application Server	Your SAP ERP or SAP S/4HANA application server

Load the definition of the /IBP/EDD_PL_FILL, /IBP/EDD_DS_DCP_FILL, /IBP/EDD_BLT_FILL, and /IBP/EDD_COR_FILL function modules there as follows:

1. Navigate to the *Tables* tab.
2. Choose *Import Objects by Name*.
3. Change the type to *Function*.
4. In the *Name* field, enter the name of the function module, and choose *OK*.
5. Repeat the previous 3 steps for the other function modules.
6. Check that the function module has been added to the list of tables.

How to Use This Template?

1. As source datastore, select the datastore that represents your SAP Integrated Business Planning system. As target datastore, select the datastore of the type File Format Group that you have previously created. Choose *Save and Define Data Flow*.
   In case you have named the file format other than **DUMMY_OUTPUT_FILE**, select the data flow, and choose *Copy to new target*. Enter a name for the data flow, select the target file format, and choose *Copy Data Flow*. Then, delete the original data flow and the target table DUMMY_OUTPUT_FILE.

2. Repeat the following steps for all data flows in the task.

3. Open the data flow for editing by choosing ▌ *Actions* ❯ *Edit* ❚.

   > **i Note**
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

4. Replace the source table with your own calculation view and recreate the mapping arrow from the source table to the QueryAndFilter transform.

5. Delete the CallRFC transform in all the data flows, replace it with a transform of type *Web Service or Function Call*, and name it **CallRFC** again.

6. Create the mapping arrows from PrepareRFC to **CallRFC** and from **CallRFC** to GetRFCOutput.

7. Choose the **CallRFC** transform. Choose *Select Function: Web Service or RFC Function* on the output node CallRFC. Select the function module you have uploaded from your SAP S/4HANA or SAP ERP datastore, and map all first-level nodes of PrepareRFC (for example, IT_PL for the IBP_DDR_to_ERP_DDPrdLoc data flow) to the corresponding nodes of CallRFC.

8. Under *Execution Properties*, you can modify the default values of the following global variables:
   - $G_LINEITEMS_PER_PACKAGE
   - $G_CURRENT_PERIOD_DATE
   - $G_PLANNING_HORIZON_IN_DAYS

   If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:
   ```
   IF (<global variable> = '<default value>')
   ```

9. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run by selecting your task and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task, and choosing ▌ *More Actions* ❯ *Promote* ❚. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.3 IBP_DDR_to_S4_Buffer_Integration_via_API

You can use the `IBP_DDR_to_S4_Buffer_Integration_via_API` template as a starting point to transfer decoupling point buffers from a demand-driven replenishment planning area in SAP Integrated Business Planning for Supply Chain (SAP IBP). This data flow serves as a template to transfer the daily buffers using top of red, top of yellow, and top of green to an SAP S/4HANA system. The horizon for transferring these buffers is set with a default of 30 days.

You can use this template together with the following templates to integrate decoupling point buffer data that is generated during the first three stages of DDMRP in SAP IBP for demand-driven replenishment:

- `IBP_DDR_to_S4_DCP_via_API`, which transfers new decoupling points or reverts them back, if necessary
- `IBP_DDR_to_S4_Position_Information`, which transfers buffer positioning information
- `IBP_DDR_to_S4_Sizing_Information`, which transfers buffer sizing information

Sharing this information with SAP S/4HANA demand-driven replenishment enables you to complete the final two stages of DDMRP there.

For more information about the components of demand-driven replenishment, search for *Integrated Demand-Driven Replenishment Process for SAP IBP and SAP S/4HANA* on the Help Portal at http://help.sap.com/ibp

The template consists of the following data flow:

Data Flow of IBP_DDR_to_S4_Buffer_Integration_via_API

Data Flow Name	Description	Source Key Figure Fields	OData V4 Service
`IBP_DDR_TO_S4_BUFFER_INTEGRATION`	Transfer top of red, top of yellow, and top of green buffers from SAP IBP for demand-driven replenishment to SAP S/4HANA for all decoupling points.	`PRDID` `LOCID` `TOPOFRED` `TOPOFYELLOW` `TOPOFGREEN` `RECDECOUPPOINTIND` `UOMID` `PERIODID0` `TSTFR` `TSTTO`	`A_ProdTimeDepdntStockLvl`

Prerequisites for Creating a Task Based on the
`IBP_DDR_to_S4_Buffer_Integration_via_API` **Template**

Creating Datastores

1. If not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading demand-driven replenishment data from the planning area in SAP IBP:
   1. Select the datastore that represents your SAP IBP system and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

   > **i Note**
   >
   > The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`. Abbreviations may occur in the name of the calculation scenario.

3. In SAP Cloud Integration for data services, create a datastore of the type OData Adapter.
   Prerequisite: You have enabled the API and set up SAP Gateway in your SAP S/4HANA system.
   Provide the following data for the new datastore:

Name	A name of your choice
Description	A description of your choice
Type	OData Adapter
Agent	The SAP Data Services Agent in your landscape
Endpoint URI	Your endpoint, for example, `/sap/opu/odata4/sap/api_prod_timedpdntstck_srv/srvd_a2x/sap/prodtimedpdntstck/0001/A_ProdTimeDepdntStockLvl`

Authentication Section	Authentication data
Default Base64 Binary Field Length	16
Depth	2
OData Version	V4
Require CSRF Header	Yes
OData Metadata Header	Minimal

4. Load the definition of the `A_ProdTimeDepdntStockLvl` OData service as follows:
   1. Navigate to the *Tables* tab.
   2. Choose *Import Objects*.
   3. Select `A_ProdTimeDepdntStockLvl` and import it.

How to Use This Template?

1. As source datastore, select the datastore that represents your SAP IBP system. As target datastore, select the datastore of the type OData Adapter that you have previously created. Choose *Save and Define Data Flow*.
2. Open the data flow for editing by choosing ▶ *Actions* ❯ *Edit* ❯.

   > **i Note**
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

3. Replace the source table with your own calculation view and recreate the mapping arrow from the source table to the `QueryAndFilter` transform.
4. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ❯ *Promote* ❯. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.4 IBP_DDR_to_S4_DCP_via_API

You can use the `IBP_DDR_to_S4_DCP_via_API` template as a starting point to transfer decoupling point buffers from a demand-driven replenishment planning area in SAP Integrated Business Planning for Supply Chain (SAP IBP). This data flow serves as a template to transfer the MRP type and initial buffer zones of your decoupling points to an SAP S/4HANA system. You can also use this template to revert the MRP type of your location products that are no longer decoupling points.

You can use this template together with the following templates to integrate decoupling point buffer data that is generated during the first three stages of DDMRP in SAP IBP for demand-driven replenishment:

- `IBP_DDR_to_S4_Buffer_Integration_via_API`, which transfers buffer zones
- `IBP_DDR_to_S4_Position_Information`, which transfers buffer position information
- `IBP_DDR_to_S4_Sizing_Information`, which transfers buffer sizing information

Sharing this information with SAP S/4HANA demand-driven replenishment enables you to complete the final two stages of DDMRP there.

For more information about the components of demand-driven replenishment, search for *Integrated Demand-Driven Replenishment Process for SAP IBP and SAP S/4HANA* on the Help Portal at http://help.sap.com/ibp

> **i Note**
>
> The `IBP_DDR_to_S4_DCP_via_API` data flow contains a delta filter logic for decoupling points. For an initial transfer, you need to change this filter. You need to run this data flow at least the same day you execute the *Recommend Decoupling Points (Solve)* planning operator. Otherwise, the delta filter will not find new or reverted decoupling points.
>
> The delta logic compares the values from yesterday, and in case you have a change of decoupling points within the same day, it is not detected, and the first change of the day remains in the system. If this is a valid use case for your business, you need to further enhance the template.

In the `IBP_DDR_TO_S4_REVERT_TO_NDCP` data flow, the MRP type is set to `PD` in the `Target_Query` transform, but you can change the default value to any specific MRP type that you use in your SAP S/4HANA system.

The template consists of the following data flows:

Data Flows of IBP_DDR_to_S4_DCP_via_API

Data Flow Name	Description	Source Key Figure Fields	OData V2 Service
IBP_DDR_TO_S4_DCP	Integrates new decoupling points from SAP IBP for demand-driven replenishment to SAP S/4HANA using the PRODUCT_SRV API.	PRDID LOCID MRPTYPE TOPOFRED TOPOFYELLOW TOPOFGREEN RECDECOUPPOINTIND DDDECOUPPPOINTCHANGEIN DICATOR PERIODID0 TSTFR	A_ProductSupplyPlannin g
IBP_DDR_TO_S4_REVERT_T O_NDCP	Changes the MRP type in SAP S/4HANA to reflect changes of reverting a decoupling point to a non-decoupling point in SAP IBP for demand-driven replenishment using the PRODUCT_SRV API.	PRDID LOCID RECDECOUPPOINTIND DDDECOUPPPOINTCHANGEIN DICATOR PERIODID0 TSTFR	A_ProductSupplyPlannin g

Prerequisites for Creating a Task Based on the IBP_DDR_to_S4_DCP_via_API Template

Creating Datastores

1. If not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning

Data Type	Value
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading demand-driven replenishment data from the planning area in SAP IBP:
   1. Select the datastore that represents your SAP IBP system and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

   > **i Note**
   >
   > The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`. Abbreviations may occur in the name of the calculation scenario.

3. In SAP Cloud Integration for data services, create a datastore of the type OData Adapter.
   Prerequisite: You have enabled the API and set up SAP Gateway in your SAP S/4HANA system.
   Provide the following data for the new datastore:

Name	A name of your choice
Description	A description of your choice
Type	OData Adapter
Agent	The SAP Data Services Agent in your landscape
Endpoint URI	Your endpoint, for example, `/sap/opu/odata/sap/API_PRODUCT_SRV`
Authentication Section	Authentication data
Default Base64 Binary Field Length	16
Depth	2
OData Version	V2
Require CSRF Header	Yes
OData Metadata Header	None

4. Load the definition of the `A_ProductSupplyPlanning` OData service as follows:
   1. Navigate to the *Tables* tab.
   2. Choose *Import Objects*.

3. Select `A_ProductSupplyPlanning` and import it.

How to Use This Template?

1. As source datastore, select the datastore that represents your SAP IBP system. As target datastore, select the datastore of the type OData Adapter that you have previously created. Choose *Save and Define Data Flow*.

2. > **i** Note
   >
   > Repeat steps 2–4 for all data flows in the task.

   Open the data flow for editing by choosing ▶ *Actions* ❯ *Edit* ❯.

   > **i** Note
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

3. Replace the source table with your own calculation view and recreate the mapping arrow from the source table to the `QueryAndFilter` transform.
4. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ❯ *Promote* ❯. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.5 IBP_DDR_to_S4_Position_Information

You can use the `IBP_DDR_to_S4_Position_Information` template as a starting point to transfer buffer position information from a demand-driven replenishment planning area in SAP Integrated Business Planning for Supply Chain (SAP IBP). This data flow serves as a template to transfer product, location, currency, and variability category information to an SAP S/4HANA system.

You can use this template together with the following templates to integrate decoupling point buffer data that is generated during the first three stages of DDMRP in SAP IBP for demand-driven replenishment:

- `IBP_DDR_to_S4_DCP_via_API`, which transfers new decoupling points or reverts them back, if necessary

- `IBP_DDR_to_S4_Buffer_Integration_via_API`, which transfers buffer zones
- `IBP_DDR_to_S4_Sizing_Information`, which transfers buffer sizing information

Sharing this information with SAP S/4HANA demand-driven replenishment enables you to complete the final two stages of DDMRP there.

For more information about the components of demand-driven replenishment, see DDMRP Supply Generation with SAP S/4HANA for Demand-Driven Replenishment.

The template consists of the following data flow:

Data Flow of IBP_DDR_to_S4_Position_Information

Data Flow Name	Description	Source Attribute Fields	Target Function Module
`IBP_DDR_to_S4_Position _Information`	Transfers new buffer position information from SAP IBP for demand-driven replenishment to SAP S/4HANA.	PRDID LOCID CURRID VARIABILITYCATEGORY	`PP_DD_IBP_POSITION`

Prerequisites for Creating a Task Based on the `IBP_DDR_to_S4_Position_Information` Template

Creating Datastores

1. If not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading demand-driven replenishment data from the planning area in SAP IBP:
   1. Select the datastore that represents your SAP IBP system and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

> **i Note**
>
> The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`. Abbreviations may occur in the name of the calculation scenario.

3. In SAP Cloud Integration for data services, create a datastore of the type *File Format Group*. This file is created because every data flow in SAP Cloud Integration for data services needs to have a target, and the function module cannot be a target. SAP IBP cannot be the target either, as it is already the source, and it cannot be both a source and a target. Therefore, the file is always created, but it does not contain any useful information.
   Prerequisite: The *File Format Type* datastore requires a root directory on the agent. You have to create the root directory, publish it on the agent, and add it to the agent's configuration before you create the following datastore.
   Provide the following data for the new datastore:

Name	A name of your choice, for example, `File_Format_Group`
Description	A description of your choice, for example, `File share for output files of data flow`
Type	*File Format Group*
Agent	The SAP Data Services Agent in your landscape
Root Directory	Name of the directory on the agent system for storing the output files, for example, `C:\HCI_Share` for a Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. In SAP Cloud Integration for data services, create a datastore for your SAP S/4HANA system.

Name	A name of your choice, for example, `SAP S/4HANA`
Description	A description of your choice, for example, `SAP S/4HANA Datastore`
Type	SAP Business Suite Applications
Agent	Your data service agent
Application Server	Your SAP S/4HANA application server

Load the definition of the `PP_DD_IBP_POSITION` function module by completing the following steps:

1. Navigate to the *Tables* tab.
2. Choose *Import Objects by Name*.
3. Change the type to *Function*.
4. In the *Name* field, enter the name of the function module, and choose *OK*.

5. Check that the function module has been added to the list of tables.

How to Use This Template?

1. Under *Projects*, open your project or create a new one.
2. Create a task in your project to transfer the buffer position data to SAP S/4HANA. Select *Use Template* and select IBP_DDR_to_S4_Position_Information as your template.
3. As your source datastore, select the datastore that represents your SAP IBP system. As your target datastore, select the datastore of the type *File Format Group* that you have previously created. Choose *Save and Define Data Flow*.
   In case you have named the file format other than **File_Format_Group**, select the data flow, and choose *Copy to new target*. Enter a name for the data flow, select the target file format, and choose *Copy Data Flow*.
4. Open the data flow for editing by choosing ❘▶ *Actions* ❯ *Edit* ❮.
5. Delete the RFC_Call transform in this data flow, replace it with a transform of type *Web Service or Function Call*, and name it **RFC_Call** again.
6. Create the mapping arrows from PPH_DD_PRODUCT to **RFC_Call** and from **RFC_Call** to Response.
7. Choose the **RFC_Call** transform. Choose *Select Function: Web Service or RFC Function* on the output node RFC_Call. Select the function module you have uploaded from your SAP S/4HANA datastore. Map all first-level nodes of PPH_DD_PRODUCT to the corresponding node of RFC_Call. Then, map ET_Message from RFC_Call to the Response transform.
8. Under *Execution Properties*, you can modify the default values of the following global variables:
   * $G_HORIZ_DAYS_PLUS
   * $G_HORIZ_DAYS_MINUS
   * $G_PQR_VALUE
   * $G_HORIZON
   If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:
   IF (<global variable> = '<default value>')
9. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run by selecting your task and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task, and choosing ❘▶ *More Actions* ❯ *Promote* ❮. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.6 IBP_DDR_to_S4_Sizing_Information

You can use the `IBP_DDR_to_S4_Sizing_Information` template as a starting point to transfer buffer sizing information from a demand-driven replenishment planning area in SAP Integrated Business Planning for Supply Chain (SAP IBP). This data flow serves as a template to transfer average daily usage, decoupled lead time, item-type category, lead-time category, lead time factor, variability category, variability factor, and minimum order quantity to an SAP S/4HANA system.

You can use this template together with the following templates to integrate decoupling point buffer data that is generated during the first three stages of DDMRP in SAP IBP for demand-driven replenishment:

- `IBP_DDR_to_S4_DCP_via_API`, which transfers new decoupling points or reverts them back, if necessary
- `IBP_DDR_to_S4_Buffer_Integration_via_API`, which transfers buffer zones
- `IBP_DDR_to_S4_Position_Information`, which transfers buffer position information

Sharing this information with SAP S/4HANA demand-driven replenishment enables you to complete the final two stages of DDMRP there.

For more information about the components of demand-driven replenishment, see DDMRP Supply Generation with SAP S/4HANA for Demand-Driven Replenishment.

The template consists of the following data flow:

Data Flow of IBP_DDR_to_S4_Sizing_Information

Data Flow Name	Description	Source Attribute and Key Figure Fields	Target Function Module
IBP_DDR_to_S4_Sizing_Information	Transfers new buffer sizing information from SAP IBP for demand-driven replenishment to SAP S/4HANA.	Attribute fields: • PRDID • LOCID • CURRID • VARIABILITYCATEGORY • ITEMTYPECATEGORY • ADUPASTHORIZON • ADUFUTUREHORIZON • SOURCETYPE • ORDERCYCLE Key figure fields: • DECOUPLEDLEADTIMEDAYS • LEADTIMEFACTORUSED • VARIABILITYFACTORUSED • MOQ • DDTOTALADU • DDDEMANDADJUSTMENTFACTOR	PP_DD_IBP_SIZING

Prerequisites for Creating a Task Based on the IBP_DDR_to_S4_Sizing_Information **Template**

Creating Datastores

1. If not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading demand-driven replenishment data from the planning area in SAP IBP:
   1. Select the datastore that represents your SAP IBP system and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

   > **i Note**
   >
   > The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`. Abbreviations may occur in the name of the calculation scenario.

3. In SAP Cloud Integration for data services, create a datastore of the type *File Format Group*. This file is created because every data flow in SAP Cloud Integration for data services needs to have a target, and the function module cannot be a target. SAP IBP cannot be the target either, as it is already the source, and it cannot be both a source and a target. Therefore, the file is always created, but it does not contain any useful information.
   Prerequisite: The *File Format Type* datastore requires a root directory on the agent. You have to create the root directory, publish it on the agent, and add it to the agent's configuration before you create the following datastore.
   Provide the following data for the new datastore:

Name	A name of your choice, for example, `File_Format_Group`
Description	A description of your choice, for example, `File share for output files of data flow`
Type	*File Format Group*
Agent	The SAP Data Services Agent in your landscape
Root Directory	Name of the directory on the agent system for storing the output files, for example, `C:\HCI_Share` for a Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. In SAP Cloud Integration for data services, create a datastore for your SAP S/4HANA system.

Name	A name of your choice, for example, `SAP S/4HANA`
Description	A description of your choice, for example, `SAP S/4HANA Datastore`

Type	SAP Business Suite Applications
Agent	Your data service agent
Application Server	Your SAP S/4HANA application server

Load the definition of the `PP_DD_IBP_SIZING` function module by completing the following steps:

1. Navigate to the *Tables* tab.
2. Choose *Import Objects by Name*.
3. Change the type to *Function*.
4. In the *Name* field, enter the name of the function module, and choose *OK*.
5. Check that the function module has been added to the list of tables.

How to Use This Template?

1. Under *Projects*, open your project or create a new one.
2. Create a task in your project to transfer the buffer position data to SAP S/4HANA. Select *Use Template* and select `IBP_DDR_to_S4_Sizing_Information` as your template.
3. As your source datastore, select the datastore that represents your SAP Integrated Business Planning system. As your target datastore, select the datastore of the type *File Format Group* that you have previously created. Choose *Save and Define Data Flow*.
   In case you have named the file format other than **File_Format_Group**, select the data flow, and choose *Copy to new target*. Enter a name for the data flow, select the target file format, and choose *Copy Data Flow*.
4. Open the data flow for editing by choosing ▌ *Actions* ❯ *Edit* ▐.
5. Delete the `RFC_Call` transform in this data flow, replace it with a transform of type *Web Service or Function Call*, and name it **RFC_Call** again.
6. Create the mapping arrows from `PPH_DD_PROD_DADJ`, `PPH_DD_PROD_DETS`, and `PPH_DD_PROD_HDR` to **RFC_Call** and from **RFC_Call** to `Response`.
7. Choose the **RFC_Call** transform. Choose *Select Function: Web Service or RFC Function* on the output node `RFC_Call`. Select the function module you have uploaded from your SAP S/4HANA datastore.
   Map all first-level nodes of to the corresponding node of `PPH_DD_PROD_DADJ`, `PPH_DD_PROD_DETS`, and `PPH_DD_PROD_HDR` to the corresponding node of `RFC_Call`. Then, map `ET_Message` from `RFC_Call` to the `Response` transform.
8. Under *Execution Properties*, you can modify the default values of the following global variables:
   * `$G_HORIZ_DAYS_PLUS`
   * `$G_HORIZ_DAYS_MINUS`
   * `$G_PQR_VALUE`
   * `$G_HORIZON`
   If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:
   ```
   IF (<global variable> = '<default value>')
   ```
9. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run by selecting your task and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ❯ *Promote* ◀. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.7 SOP_KF_Actuals

This is a task for extracting actuals or shipment history key figure data from SAP ERP, and loading it to SAP Integrated Business Planning for sales and operations.

Source extractor is `2LIS_12_VCITM_SOP`, source tables are `VBRK`, and `VBRP`. The task loads key figure data into the `SOPDD_STAGING_KFTAB_SM1BASE` staging table, into the `SM1ACTUALSQTY` and `SM1ACTUALSREV` key figure columns.

Prerequisites for Creating a Task Based on the SOP_KF_Actuals Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the actuals or shipment history key figure data from SAP ERP. Select *Use Template*, and select `SOP_KF_Actuals` as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.4.8 SOP_KF_Inventory

This is a task for extracting current inventory and inventory target key figure data from SAP ERP, and loading it to SAP Integrated Business Planning for sales and operations.

Source tables are `MARC`, `MARA`, and `MARD`. The task loads key figure data into the `SOPDD_STAGING_KFTAB_SM1BASE` staging table, into the `SM1INITIALINVENTORYQTY` and SM1INVENTORYTARGETQTY key figure columns.

Prerequisites for Creating a Task Based on the SOP_KF_Inventory Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the inventory key figure data from SAP ERP. Select *Use Template*, and select `SOP_KF_Inventory` as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.4.9 SOP_KF_OpenOrders

This is a task for extracting open orders key figure data from SAP ERP, and loading it to SAP Integrated Business Planning for sales and operations.

Source extractors are `2LIS_11_VAHDR`, `2LIS_11_VAITM`, and `2LIS_11_VASTI`.

The task loads key figure data into the `SOPDD_STAGING_KFTAB_SM1BASE` staging table, into the `SM1INITIALINVENTORYQTY` and `SM1INVENTORYTARGETQTY` key figure columns.

Prerequisites for Creating a Task Based on the SOP_KF_OpenOrders Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the open orders key figure data from SAP ERP. Select *Use Template*, and select `SOP_KF_OpenOrders` as template.

3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.4.10 SOP_KF_SalesForecastPrice

This is a task for extracting sales forecast price key figure data from SAP ERP, and loading it to SAP Integrated Business Planning for sales and operations.

Source extractor is `2LIS_13_VDITM`.

The task loads key figure data into the `SOPDD_STAGING_KFTAB_SM1BASE` staging table, into the `SM1SALESFORECASTPRICE` key figure column.

Prerequisites for Creating a Task Based on the SOP_KF_SalesForecastPrice Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the sales forecast price key figure data from SAP ERP. Select *Use Template*, and select `SOP_KF_SalesForecastPrice` as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.4.11 SOP_KF_CapacityLimit

This is a task for extracting capacity limit or available resource capacity key figure data from SAP APO, and loading it to SAP Integrated Business Planning for sales and operations.

Source table is `/SAPAPO/RES_HEAD`, source extractor is `9ACAPACITY`.

The task loads key figure data into the `SOPDD_STAGING_KFTAB_SM1BASE` staging table, into the `SM1CAPASUPPLY` key figure column.

Prerequisites for Creating a Task Based on the SOP_KF_CapacityLimit Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the capacity limit or available resource capacity key figure data from SAP ERP. Select *Use Template*, and select `SOP_KF_CapacityLimit` as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.
5. Review the default values of the global variables, and make changes as needed.

7.4.12 SOP_KF_Consumption

This is a task for extracting capacity consumption rate key figure data from SAP APO, and loading it to SAP Integrated Business Planning for sales and operations.

Source BAPI functions are `BAPI_LOCSRVAPS_GET_LIST2` and `BAPI_PDSSRVAPS_GETLIST`.

The task loads master data into the `SOPDD_STAGING_KFTAB_SM1BASE` staging table, into the `SM1CAPACONSUMPTION` key figure column.

Prerequisites for Creating a Task Based on the SOP_KF_Consumption Template

The source and target datastores must exist.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the capacity consumption rate key figure data from SAP ERP. Select *Use Template*, and select `SOP_KF_Consumption` as template.
3. In your newly created task, select the data flow, and choose *Copy to New Target*. Enter a name for the data flow. Select the target table, and choose *Copy Data Flow*.
4. Modify your data flow, if needed, to specify the field mappings.

5. Review the default values of the global variables, and make changes as needed.

7.4.13 IBP_KF_DemandPlanning

The `IBP_KF_DemandPlanning` template contains a data flow to extract demand planning-related key figure data, such as the consensus demand plan data, from SAP APO, and load it to SAP Integrated Business Planning for demand or for sales and operations.

> → Recommendation
>
> When defining the data transfer, take the following recommendations into consideration:
>
> SAP recommends transferring demand planning key figure data from SAP APO at product-location-customer level. In case the customer characteristic is not available for the key figure data in the source system, either use a similar characteristic, such as the customer group, or assign a constant value to the customer in SAP Cloud Integration for data services.
>
> The template uses weeks are periods, which is the recommended period to use. You can also use months.

Prerequisites for Creating a Task Based on the IBP_KF_DemandPlanning Template

Defining and Releasing DataSources in SAP APO

In SAP APO, you have executed the following steps:

1. You have set up your planning area in SAP APO.
2. You have generated the necessary DataSources (in the *DP/SNP Data Extraction* (`/SAPAPO/SDP_EXTR`) transaction).
   Generate a DataSource for each key figure that you want to upload from SAP APO. For performance reasons, consider using parallel processing. Specify a parallel processing profile to do so.
   In your key figure DataSource, select product, location, customer, planning version, time period (weeks recommended), and the relevant key figure, such as consensus forecast.
   Make sure you have hidden all fields that you do not need to extract.
3. Replicate each DataSource without transporting it.
4. In the Data Warehousing Workbench, activate the DataSources.
5. Execute the *Release 9A DataSources for External Use* (`/SAPAPO/PAREA_EXTR_EXPOSE`) report to release your DataSources for external use, that is, to make them available for SAP Cloud Integration for data services.

   > i Note
   >
   > In case the `/SAPAPO/PAREA_EXTR_EXPOSE` report is not available in your system, implement SAP Note 2085981 .

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

Prerequisite: SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds .

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP APO system. Provide the following data for the new datastore:

Name and Description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application server	
Authentication and credentials or SNC details	
System number and client number of your SAP APO / SAP SCM system	

2. Import the DataSources that you have created in SAP APO. Select the datastore that represents your SAP APO / SAP SCM system, and choose *Import Object by Name*. Select *Extractors*, and enter the name of the DataSource in the *Name* field.

3. Create a datastore that represents your SAP Integrated Business Planning system. Provide the following data for the new datastore:

Name and Description	Instance
Type	Integrated Business Planning
Instance	Your SAP Integrated Business Planning system

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

4. Import the staging tables that will be used for importing key figure data:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Timeseries folder* in the tree, and select the planning area from which you will import the data.
   3. Expand the planning area, select the staging table, and choose *Import*.

   > **i Note**
   >
   > The name of the staging table is of the following pattern: `SAPMD_STAGING_KFTAB_<Planning Area Version><Planning Area Name>`.
   >
   > Abbreviations may occur in the name of the table.

> Make sure to select tables that have this naming convention.

How to Use This Template?

1. Under *Projects*, create your own project to group your data integration tasks, or select an existing project.
2. Create a task in your project to transfer the demand planning-related key figure data from your planning area in SAP APO. Select *Use Template*, and select `IBP_KF_DemandPlanning` as template.
3. As source datastore, select the datastore that represents your SAP APO / SAP SCM system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save* and *Define Data Flow*.
   You can find the `SOPDD_STAGING_KFTAB_SAP6SAP6` target object (tables) and the related `DF_IBP_DemandPlanning` data flow under the task you have created.
4. If you want to transfer data from a different planning area than the one in the template, replace the target table of the data flow. To do this, select the data flow, and choose *Copy to new target*.
5. Enter a name for the data flow, and select the staging table of your planning area as target object. If you have not imported the staging table for your planning area to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Open the data flow for editing by choosing ▶ *Actions* ❯ *Edit* ◀.

   > **i Note**
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs.
   >
   > This document provides you with more detailed information about the required adjustments.

7. Replace the source table with your DataSource for the key figure (the DataSource you have generated and exposed in SAP APO).
8. Connect the source table to the `Read_KeyFigure` step.
9. In the `Read_KeyFigure` step, define the field mappings for calendar week, customer, product, location, and planning version.
   Open the field mapping for `MATID`, and replace the *APO* string in lookup ('`APO./SAPAPO/MATKEY`' with the name of your source datastore.
   Replace the string *"9ADP_CUSTOM_TOTAL_DEMAND_PLAN"."/BIO/9AMATNR"* with the product ID field in your source table.
10. In the `Lookup_ConvFactors` step, open the mapping for the fields `NUMERATOR` and `DENOMINATOR`, and replace the *APO* string in lookup ('`APO./SAPAPO/MARM`' with the name of your source datastore.
11. In the `MapToTarget` step, define the mappings for the key figures you want to transfer.
12. Make sure you have mapped the date and at least one key figure. In case you have mapped several key figures, they must refer to the same planning level. Check that you have mapped all key fields of the planning level.
13. Once you have adjusted each data flow to your needs, set the following global variables under *Execution Properties*:
    - `$G_PLANNING_AREA`
    - `$G_PLANNING_UOM`

    If you want to transfer key figure values at an aggregated time level, specify the time profile level for upload in the `$G_TIME_PROFILE_LEVEL` variable. In this case, the key figure values will be automatically

disaggregated to their base planning level. You find the numeric ID of the time profile level on the *Time Profiles* screen in the *Configuration* app.

If you don't specify a value for the `$G_TIME_PROFILE_LEVEL` parameter, key figure values are loaded at their base planning level.

> **i Note**
>
> You can restrict the upload of key figure values to the upload at the base planning level only. To do so, enter **x** or **X** as the value of the `KF_UPLOAD_EXT_AT_BASE_LEVEL_ONLY` global configuration parameter in the *Configuration* app.

14. Delete the template data flow and its associated target table from your task.
15. Choose *Validate* to check your task. If required, make the necessary corrections.

> **i Note**
>
> By default, the template uses weeks as periods. In case you want to use months, make the following adjustments:
>
> 1. Set up your DataSource in SAP APO to extract data at monthly level.
> 2. In the `Read_KeyFigure` step in the dataflow of your task, add `CALMONTH` as an output field, and map it to the `CALMONTH` input field. Delete the `CALWEEK` field from the output structure.
> 3. In the `Lookup_ConvFactors` step, add `CALMONTH` as an output field, and map it to the `CALMONTH` input field. Delete the `CALWEEK` field from the output structure.
> 4. In the `Calc_KeyFigure` step, add `CALMONTH` as an output field, and map it to the `CALMONTH` input field. Delete the `CALWEEK` field from the output structure.
>   Select `KEYFIGUREDATE`, and edit the *Mapping* under *Transform Details*: Comment the `# Transfer CALWEEKS` section, and uncomment the `# Transfer CALWEEKS` section.

Executing the Task

Make sure you have imported the necessary master data before you execute the task to load demand planning-related data into the demand or sales and operations application.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ❯ *Promote* ❯. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

> **i Note**
>
> Between SAP APO and SAP Integrated Business Planning, the data transfer is always a complete data transfer, not a delta transfer.
>
> Deletions of characteristic value combinations in SAP APO's Demand Planning cannot be transferred to SAP Integrated Business Planning.

7.4.14 IBP_KF_PlannedIndependentRequirements

You can use the `IBP_KF_PlannedIndependentRequirements` template as a starting point to transfer sensed demand data (short-term forecasting data created by demand sensing) from SAP Integrated Business planning for demand to SAP APO Supply Network Planning as planned independent requirements.

You can transfer sensed demand data at product-location level only. This template uses days as time periods.

Prerequisites for Creating a Task Based on the IBP_KF_PlannedIndependentRequirements Template

Defining a Web Service in SAP APO

You have to define a Web service in your SAP SCM/SAP APO system in order to receive sensed demand data from SAP Integrated Business Planning.

1. In your SAP SCM/SAP APO system, start the *Function Builder* (`SE37`) transaction, and generate a Web service from the `BAPI_PIRSRVAPS_SAVEMULTI` function module.
   In the *SOAP Application* field, do not change the default value. In the *Profile* field, choose *Authentication with User and Password, No Transport Guarantee*.
   Save and activate your Web service.
2. Call the SOA Management (`SOAMANAGER`) transaction. Under *Service Administration*, choose *Web Service Configuration*.
   Select the Web Service you have created in `SE37`. Choose *Create Service*. Enter a name and description for the Web service, and a name for the binding.
   Make sure that you select *SSL* for *Transport Level Security*, and *User ID/Password* for *Transport Channel Authentication*.
3. Generate the URL for your Web service. In the SOA Management (`SOAMANAGER`) transaction, open your Web service, and in the line that represents the binding, click the *"Open Binding WSDL Generation"* icon.
   In the *WDSL Generation for Binding* screen, make sure that you select *Alternative URL*, and *https* as *Meta Data Protocol*. Choose *Apply URL Options*.
   The URL, that you can use when you create a datastore in SAP Cloud Integration for data services is displayed.

For more information about generating Web services, see the documentation of Application Server ABAP in the SAP NetWeaver Library for your release at http://help.sap.com/nw.

Creating Datastores

1. Create a datastore - if it is not yet available - that represents your SAP Integrated Business Planning system. Provide the following data for the new datastore:

Name and Description

Type	Integrated Business Planning
Instance	Your SAP Integrated Business Planning system

> **i Note**
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

2. Import the calculation scenario that will be used for **reading demand sensing data** from the planning area in SAP Integrated Business Planning:
   1. Select the datastore that represents your SAP Integrated Business Planning system, and choose *Import Objects*.
   2. Expand the *Calculation Scenario* folder in the tree, and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

   > **i Note**
   >
   > The name of the calculation scenario is of the following pattern: `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`.
   >
   > Abbreviations may occur in the name of the calculation scenario.

3. In SAP Cloud Integration for data services, create a datastore for the Web service call to SAP APO. Provide the following data for the new datastore:

Name and Description

Type	SOAP Web Service
Agent	The SAP Data Services Agent in your landscape
WDSL Path	The URL of the Web service
User name	
Password	

On the *Tables* tab, choose *Import Objects*. Select and import the signature of the generated Web service.

4. In SAP Cloud Integration for data services, create a datastore of the type File Format Group.
   Prerequisite: The File Format Type datastore requires a root directory on the agent. You have to create the root directory, publish it on the agent, and add it to the agent's configuration before you create the following datastore.
   Provide the following data for the new datastore:

Name and Description

Type	File Format Group
Agent	The SAP Data Services Agent in your landscape
Root directory	

On the *File Formats* tab, choose ▌▶ *Create File Format* ▶ *Create XML Schema* ▌.

Create a new XML schema by uploading a file that has a similar structure:

```xml
<?xml version="1.0" encoding="ASCII"?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ResponseOfBAPI_PIRSRVAPS_SAVEMULTI">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SuccessMessage">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Item" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="FilledOnSuccess"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

How to Use This Template?

1. Under *Projects*, open your project. You can also create a new project for sending the demand sensing data from SAP Integrated Business Planning to Supply Network Planning (SNP) in SAP APO.
2. Create a task in your project to transfer the demand sensing data to Supply Network Planning. Select *Use Template*, and select `IBP_KF_PlannedIndependentRequirements` as template.
3. As source datastore, select the datastore that represents your SAP Integrated Business Planning system. As target datastore, select the datastore of the type File Format Group that you have previously created. Choose *Save and Define Data Flow*.
   You can find the `APO_WS_RESPONSE` target object (table) and the related DF_IBP_SensedDemand data flow under the task you have created.
4. Select the data flow, and choose *Copy to new target*. Enter a name for the data flow, and select the target table, then choose *Copy Data Flow*.
5. Open the data flow for editing by choosing ▶ *Actions* ❭ *Edit* ❭.

   > **i Note**
   >
   > In the note box of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

6. Replace the source table with your own calculation view.
7. Connect the source table to the `Read_Calc_View` step.
8. Adjust the mappings of the output fields to the input fields in the `Read_Calc_View` step.
9. Delete the `WebService_Call` step.

10. Insert a new Web service step, and rename it to `WebService_Call`. Connect the `WebService_Call` step to the `WebService_Payload` step (step before the `WebService_Call` step), and to the `Read_WebService_Response` step (this is the step after the `WebService_Call` step).
11. Open the `WebService_Call` step. In the *Output* pane, choose *Select Web Service Function*. Select the structure under your Web service datastore, and choose *OK*.
12. Map the `BAPI_PIRSRVAPS_SAVEMULTI` output node to the input node of the same name.
13. Check, and if needed, adjust the mapping and the iteration rules in the `Read_WebService_Response` step.
14. A filter in the `Print_WebService_Response` step specifies which type of messages will be written to the log.
    The template prints only error messages (type 'E') to the log. In case you want warnings and information messages as well to appear in the log, add the respective types ('W' and 'I') to the filter of the `TYPE` field.
15. Check, and if needed, adjust the mapping and the iteration rules in the `Target_Mapping` step.
16. Under *Execution Properties*, set the global variables `$G_LOGICAL_SYSTEM` and `$G_PLANNING_UOM`. Check, and if needed, set the `$G_PLANNING_VERSION`, `$G_DAYS_TO_TRANSFER`, and `$G_LINEITEMS_PER_PACKAGE`, `$G_ATP_CATEGORY`, and `$G_FROM_DAYS_IN_FUTURE` global variables.
17. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▌ *More Actions* ❯ *Promote* ▐. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

Parallel Web Service Calls

In order to improve performance, you can configure a task to execute the Web service in parallel mode. To do this, select the *Enable parallel execution* checkbox on the *Properties* tab of your Web service datastore.

In the data flow definition, specify the degree of parallelization.

1. Select your data flow, and choose *Edit*.
2. Choose *View Options* (the first icon next to the data flow name).
3. Enter a value in the *Degree of parallelism* field to specify the number of parallel threads.

> **i Note**
>
> If you enter **default**, the system takes the value from the `Global_DOP` parameter from the `DSconfig.txt` configuration file on the SAP Data Services Agent.
>
> We recommend that you do not change the degree of parallelism at a global level (in the `DSconfig.txt` file). Instead, specify it for each data flow where needed, by entering a positive integer in the *Degree of parallelism* field.

> ⚠ **Caution**
>
> Locking in SAP APO takes place at the level of product and location combination. Therefore, all line items of a certain product and location must be in the same Web service call to avoid locking conflicts or time-out of the Web service call. This is guaranteed by the packaging logic modeled in the template. We recommend that you do not change this part of the template.

By default, SAP Cloud Integration for data services distributes the workload to the parallel threads in groups of 50 Web service calls. In case there are 50 or less Web service calls, parallel processing does not take place.

The minimum number of Web service calls required for parallelization may be different in SAP Cloud Integration for data services.

The number of Web service calls is calculated by dividing the total number of line items by the line items per package, For example, there are 125.000 line items to be transferred, and 1.000 line items form a package. In this case, 125.000/ 1.000 = 125 Web service call are needed.

You will reach the best performance if every thread executes the same number of Web service calls (which is a multiple of 50). You can influence the number of Web service calls by adjusting the number of line items per package (global variable `$G_LINEITEMS_PER_PACKAGE`).

> **i Note**
>
> When using parallel processing, the memory of the SAP Data Services Agent might become a bottleneck. You can mitigate this by using smaller package sizes when you increase the degree of parallelization (for example, doubling the number of threads and cutting the package size by half would lead to a similar memory consumption).

When you define the number of threads and the package size, consider the following example.

> **❖ Example**
>
> There are 125 Web service calls to be executed, and the SAP Data Services has been configured to use 4 parallel threads. The 125 Web service calls are distributed as follows:
>
> | N/A | Thread 1 | Thread 2 | Thread 3 | Thread 4 |
> | --- | --- | --- | --- | --- |
> | Number of Web service calls | 50 | 50 | 25 | 0 |
>
> In case the total number of Web service calls is 50 or lower, no parallel execution takes place.
>
> As the total number of Web service calls depends on the number of selected line items and the package size, choose a package size that provides a good utilization of parallel processing.

7.4.15 IBP_KF_POSSales and IBP_KF_POSStock

The `IBP_KF_POSSales` template contains a data flow to extract POS sales data from SAP Demand Signal Management (SAP DSiM) and load it to a demand planning area of SAP Integrated Business Planning. This data includes retailer store sales, retailer promotional sales, and withdrawals from retailer distribution centers.

The `IBP_KF_POSStock` template contains a data flow to extract POS stock data from SAP Demand Signal Management and load it to a demand planning area of SAP Integrated Business Planning. This data covers retailer store stock and retailer distribution center stock.

Prerequisites for Creating a Task based on Template IBP_KF_POSSales or IBP_KF_POSStock

Your SAP DSiM system must be on SAP Demand Signal Management 2.0 FP4 or higher. Your SAP Cloud Integration for data services instance and your Data Services Agent must be on version 1.0.9 or higher. Your IBP system must be on version IBPCLOUD 160 or higher.

Only released POS data is transferred from SAP DSiM to IBP. Releasing POS data takes place at the level of the IBP customer, IBP manufacturer DC, and IBP week.

Therefore, you must perform the following steps:

Setting up Locations and Week Definitions for IBP in SAP DSiM

SAP transfers POS data from SAP DSiM at product-location-customer level. In SAP DSiM you have to upload the assignments of locations and customers from IBP to SAP DSiM locations before you start using the SAP Cloud Integration for data services template.

The template uses weeks as periods. Therefore, you have to define the time profile's week definition of your target planning area in SAP DSiM and to ensure that the system aggregates the POS data appropriately.

For more information about the exact steps for preparing POS data in SAP DSiM to be released for IBP (including the setup of quantity conversions in SAP DSiM), see Configuring the Integration with IBP on SAP Help Portal at http://http://help.sap.com/dsim.

Creating an ODP Source System in SAP DSiM

In SAP DSiM, you have to make the POS data available as operational data providers (ODPs). To do so, execute the following steps:

- Create an ODP source system, as described in the chapter Creating an ODP Source System on SAP Help Portal at http://help.sap.com/saphelp_nw75. Choose the ODP folder *ODP-BW*. The source system type you enter must start with an **S**. You can enter a release of your choice.
- Replicate the DataSources of the new ODP source system. By default, all DSOs and InfoObjects used as InfoProviders are listed. Deselect all entries except for the DataSources *DS91_F* and *DS92_F* and confirm. By doing so, you ensure that the DSOs /DDF/DS91 and /DDF/DS92 are exposed as ODPs. The DSOs store the POS data that has been released to IBP.

Creating Datastores and Importing DataSources and Tables in SAP Cloud Integration for data services

Prerequisite: SAP Data Services Agent is configured in your system landscape.

For more information about the SAP Data Services Agent, see the SAP Data Services Agent Guide on SAP Help Portal at http://help.sap.com/cpi_ds.

1. In SAP Cloud Integration for data services, create a datastore that represents your SAP DSiM system. Provide the following data for the new datastore:

Name and Description	
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application Server	
Authentication and credentials or SNC details	
System number and client number of your SAP DSiM system	
ODP Context	BW

2. Import the DataSources of the ODP context *BW* that you have created in SAP DSiM, as described in the previous step. Select the datastore that represents your SAP DSiM system, and choose *Import Object by Name*. Select *Extractors*, extraction mode *Change Data Capture*, and enter the name of the ODP DataSource in the *Name* field.
   For the DSOs /DDF/DS91 and /DDF/DS92, enter **/DDF/DS91$F** and **/DDF/DS92$F** as names of the extractors.

3. Create a datastore that represents your SAP Integrated Business Planning system. Provide the following data for the new datastore:

Name and Description	Instance
Type	Integrated Business Planning
Instance	Your SAP Integrated Business Planning system

> i Note
>
> If you cannot select your SAP Integrated Business Planning system as an instance, contact SAP.

4. Import the staging tables that will be used for importing key figure data. It is recommended that you use a copy of SAP Sample SAP6. This sample contains key figures that can be used as target of the key figures provided by the DataSources of your datastore.
5. Select the datastore that represents your SAP Integrated Business Planning system and choose *Import Objects*.
6. Expand the *Timeseries* folder in the tree, and select the planning area from which you will import the data.
7. Expand the planning area, select the staging table, and choose *Import*.

> i Note
>
> The name of the staging table has the following pattern: SOPDD_STAGING_KFTAB_<Planning Set Name>. In case you have copied your planning area from an SAP sample, the planning set is a concatenation of the SAP sample name and the name of your planning area. In case you use version

> data as a staging table, the planning set name is a concatenation of the version and the planning area name. Abbreviations may occur in the name of the table. Make sure to select tables that follow this naming convention.

How to Use These Templates?

1. Under *Projects*, create your own project to group your data integration tasks or select an existing project.
2. Create a task in your project to transfer the sales key figures from SAP DSiM to IBP. Select *Use Template* and select *IBP KF POSSales* as a template. Create another task in your project to transfer the stock key figures from SAP DSiM to IBP. Select *Use Template* and select *IBP_KF_POSStock*.
3. As source datastore, select the datastore that represents your SAP DSiM system. As target datastore, select the datastore that represents SAP Integrated Business Planning. Choose *Save* and *Define Data Flow*. You can find the target object (table) SOPDD_STAGING_KFTAB_SAP6 and the related data flows DF_IBP_KF_POSSales or DF_IBP_KF_POSStock under the task you have created.
4. If you want to transfer data from a different planning area than the one used in the template, replace the target table of the data flow. To do so, select the data flow, and choose *Copy to New Target*.
5. Enter a name for the data flow and select the staging table of your planning area as target object. If you have not imported the staging table for your planning area to the datastore yet, you can also perform the import in this step. Select *Copy Data Flow* or *Import and Copy Data Flow*.
6. Open the data flow for editing by choosing *Edit*.

   > **i Note**
   >
   > In the note box next to the data flow, you find instructions on how to adjust the data flow to fit your needs.

7. If you use your own key figures instead of the new key figures of SAP sample SAP6, adjust the mappings in the step *MapToTarget* accordingly.
8. Make sure you have mapped the date and at least one key figure. If you have mapped several key figures, they must refer to the same planning level. Check that you have mapped all key fields of the planning level.
9. Once you have adapted each data flow to your needs, set the global variables $G_PLANNING_AREA and $G_PLANNING_SET under *Execution Properties*.
10. Delete the template data flow and its associated target table from your task.
11. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

Ensure that you have imported the reuired master data before you execute the task to load POS data to IBP.

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task and choosing *View History*.

After a thorough test of your task, you can decide to schedule it for a periodic run. To do so, first promote your task to the production environment by selecting the task and choosing ▌ *More Actions* ❯ *Promote* ▌. Then switch to the productive view. There, select your task, choose ▌ *Schedule* ❯ *New* ▌ and enter the required data.

In addition, schedule the BW process chains in your SAP DSiM system that prepare the POS data for transfer to IBP. It is recommended that you schedule these process chains in a way that they finish well in advance of the regular start date of your corresponding task in the SAP Cloud Integration for data services.

> **i Note**
>
> The tasks for transferring POS data from SAP DSiM to IBP always execute a delta transfer of all combinations of IBP customer, manufacturer DC, and IBP week that have been changed since the last delta transfer. For each combination, the system transfers a complete image. Logical deletions of certain combinations which may occur by reassigning all the DSiM source locations of those combinations cannot be transferred to IBP.

7.4.16 IBP_MRO_SAP9_KF_S4_ERP_AddOn

The `IBP_MRO_SAP9_KF_S4_ERP_AddOn` task template contains data flows to extract Maintenance, Repair, and Operations (MRO)-relevant key figures from SAP S/4HANA or SAP ECC and to load them to SAP Integrated Business Planning for Supply Chain (SAP IBP). It uses one of the following add-ons for extraction:

- SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning. For more information about the add-on, see https://help.sap.com/ibp_s4hana_addon.
- SAP ERP, supply chain integration add-on for SAP Integrated Business Planning. For more information about the add-on, see https://help.sap.com/ibp_erp_addon.

> **⚠ Caution**
>
> The `MR9_KF_PartReplenishmentDays` data flow reads directly from a database table. Please ensure that this type of access is covered by your database license.

Data Flows of the IBP_MRO_SAP9_KF_S4_ERP_AddOn Task

Target Key Figure in SAP IBP	Data Flow Name	Data Flow Description
MROPMDEMANDQTY	MR9_KF_PrevMaintDemand	Reads MRO preventive maintenance plan orders data via the `/IBP/ MRO_MAINTPLAN_KF` extractor, and transfers them to an MRO planning area as preventive maintenance demand quantity data.
MROPMACTUALSQTY	MR9_KF_ActualsPrevMaint	Reads MRO completed maintenance plan orders data via the `/IBP/ MRO_MAINTPLAN_KF` extractor, and transfers them to an MRO planning area as preventive maintenance actuals quantity data.

Target Key Figure in SAP IBP	Data Flow Name	Data Flow Description
MINPRODUCTION	MR9_KF_MinProduction	Reads MRO called maintenance plan orders data via the `/IBP/ MRO_MAINTPLAN_KF` extractor, and transfers them to an MRO planning area as minimum production receipts data.
MROPARTUSAGEHISTORYCM	MR9_KF_ActualsComponentUsage	Reads MRO material consumption history for corrective maintenance orders data via the `/IBP/ MRO_MATCONS_CORRMAINT_KF` extractor and transfers them to an MRO planning area as actuals usage of component (corrective maintenance) data.
MROPCAPAUSAGEHISTORYCM	MR9_KF_ActualsResourceUsage	Reads MRO resource consumption history for corrective maintenance orders data via the `/IBP/ MRO_RESCONSCORRMAINT_KF` extractor, and transfers them to an MRO planning area as actuals usage of resource (corrective maintenance) data.
MRORESCOSTPERHOUR	MR9_KF_ResourceCostRate	Reads MRO resource cost per hour data via the `/IBP/ MRO_RESCOSTPERHOUR_KF` extractor, and transfers them to an MRO planning area.
COSTPERUNIT	MR9_KF_CostPerUnit	Reads MRO materials cost rate data via the `/IBP/ MRO_MaterialCostRate_KF` extractor, and transfers them to an MRO planning area as component cost rate data.
INITIALINVENTORY	MR9_KF_InitialInventory	Reads the initial *Initial Inventory* key figure with aggregation from the `/IBP/STOCK_KF` extractor, and transfers it to an MRO planning area.
MINTRANSPORT	MR9_KF_MinTransport	Reads the *MinTransport* key figure with aggregation from the `/IBP/ ORDER_KF` extractor and transfers it to an MRO planning area.
ERPSAFETYSTOCK	MR9_KF_SafetyStock	Reads the current ERP part safety stock value that is specified in the *Material Master* record from the extractor `/IBP/ LOCATIONPRODUCT_ATTR` and transfers it to an MRO planning area.

Target Key Figure in SAP IBP	Data Flow Name	Data Flow Description
CAPASUPPLY	MR9_KF_CapacitySupply	Reads the capacity supply value in seconds from the extractor `/IBP/CAPASUPPLY_KF` at an aggregated weekly time profile level, then divides it by 60 to convert the value from seconds to minutes, then again divides it by 60 to convert the value from minutes to hours, to get the available capacity per work center in hours per week. Then transfers it to an MRO planning area.
MROPARTREPLENISHMENTLEADTIMEDAYS	MR9_KF_PartReplenishmentDays	Reads the planned delivery time for the material from the vendor from the table `/IBP/TLANE_EXT` and transfers it to an MRO planning area.

7.4.17 Key Figure Templates for Production Planning Integration

You can use templates to integrate data with SAP Cloud Integration for data services for using SAP Integrated Business Planning for Supply Chain (SAP IBP) time-series-based supply planning in combination with Advanced Planning (Production Planning and Detailed Scheduling (PP/DS)) in SAP S/4HANA.

To integrate data between SAP Integrated Business Planning for Supply Chain (SAP IBP) and PP/DS, create data flows based on the following templates:

Related Information

Production Planning Integration

7.4.17.1 IBP_PPDS_from_S4_Orders

You can use the `IBP_PPDS_from_S4_Orders` template as a starting point to transfer the quantity of orders (procurement receipts, production receipts, and stock transfer requisitions (STRs)) per period from Advanced

Planning (Production Planning and Detailed Scheduling (PP/DS)) in SAP S/4HANA to SAP Integrated Business Planning for Supply Chain (SAP IBP) time-series-based supply planning into key figures *Confirmed External Receipts*, *Confirmed Production Receipts*, and *Confirmed Transport Receipts*.

Based on the ATP category of an element in PP/DS, you can decide if it should be transferred back into the confirmed key figures. An example of different ATP categories for each of the data flows is given in the templates. Revise the ATP categories and change the global variables and associated scripts based on your business requirements. You can find instructions on adjusting the data flow to fit your needs in the note boxes of the data flows.

As the ATP categories for stock transfer requisition and procurement receipts partially overlap (except for the ATP categories of subcontracting orders, which are defined only for STRs), the parameter *Location Type* is used to differentiate these orders. For the data flow writing into the key figure *Confirmed Transport Receipts*, the location types 1001 (production plant), 1002 (distribution center) and 1050 (subcontractor; but also used for the process of subcontracting without source location) are used in the template. For the data flow writing into the key figure *Confirmed External Receipts*, the location type 1021 (supplier) is used. Note that you need to revise these settings based on your business needs.

The called function module is using a propagation range to select the data. You need to define the propagation range in PP/DS for the relevant products and maintain the propagation range as a global variable in the tasks. The propagation range can be maintained using the global variable `$G_PROPAGATION_RANGE`.

> **i Note**
>
> As a source table, the flat file created using the task `IBP_PPDS_to_file_IBP_TimeProfile` is used. Therefore, it is a prerequisite to run this data flow before executing the task `IBP_PPDS_from_S4_Orders`.

The template consists of the following data flows:

Data Flow Name	Description	Source Function Module and Table	Target Key Figure
PPDS_IBP_ProcReceipts	PP/DS procurement receipts are written into the key figure CONFIRMEDRECEIPT in SAP IBP.	/SAPAPO/ SYPP_AGGREGATE_ORD	CONFIRMEDRECEIPT
PPDS_IBP_ProdOrders	PP/DS production orders are written into the key figure CONFIRMEDPRODUCTION in SAP IBP.	/SAPAPO/ SYPP_AGGREGATE_ORD	CONFIRMEDPRODUCTION
PPDS_IBP_STR	PP/DS STRs are written into the key figure CONFIRMEDTRANSPORT in SAP IBP.	/SAPAPO/ SYPP_AGGREGATE_ORD	CONFIRMEDTRANSPORT

Prerequisites for Creating a Task Based on the IBP_PPDS_from_S4_Orders Template

Creating Datastores

1. If it is not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP IBP system as an instance, contact SAP.

2. Import the SAP IBP staging tables for key figures of the target planning area as follows:
   1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
   2. Expand the *Timeseries Folder* in the tree and select your planning area.
   3. Expand the planning area, select the staging table, and choose *Import*.

   > **i Note**
   >
   > The name of the staging table follows the pattern `SOPDD_STAGING_KFTAB_<Example Planning Area><Planning Area>` (do not use the table with the suffix `_REP`).
   >
   > Abbreviations may occur in the name of the calculation scenario.

3. If not yet available, create a datastore for your SAP S/4HANA system with the following data:

Name	A name of your choice, for example, **SAPS4HANA**
Description	A description of your choice, for example, **SAP S/4HANA Datastore**
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application Server	Your SAP S/4HANA application server

4. Load the definition of the `/SAPAPO/SYPP_AGGREGATE_ORD` function module there as follows:
   1. Navigate to the *Tables* tab.
   2. Choose *Import Objects by Name*.
   3. Change the type to *Function*.

4. In the *Name* field, enter the name of the function module, and choose *OK*.
5. Check that the function module has been added to the list of tables.

> **i Note**
>
> If the datastore has several configurations, make sure the configuration of the system for loading the function module is the default one. If it isn't, make it the default and don't forget to save the changes.

How to Use This Template?

1. As source datastore, select the file format group datastore you created for the `IBP_PPDS_to_file_IBP_TimeProfile` task.
2. As target datastore, select the datastore that represents your SAP IBP system.
3. Open the data flow for editing by choosing ▶ *Actions* ❯ *Edit* ◗.

> **i Note**
>
> In the note boxes of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

4. Check the definition `QueryAndFilterLocType` (where available) and, if necessary, adapt it to your needs.
5. Delete the `CallRFC` transform in all the data flows, replace it with a transform of type *Web Service or Function Call*, and name it `CallRFC` again.
6. Create the mapping arrows from `PrepareRFC` to `CallRFC` and from `CallRFC` to `GetRFCOutput`.
7. Choose the `CallRFC` transform. Choose *Select Function: Web Service or RFC Function* on the output node `CallRFC`. Select the function module you have uploaded from your SAP S/4HANA datastore, and map all first-level nodes of `PrepareRFC` (for example, `IT_ATPCAT`) to the corresponding nodes of `CallRFC`.
8. Under *Execution Properties*, you can modify the default values of the following global variables:

Global Variable	Data Type	Default Value	Description
`$G_PROPAGATION_RANGE`	VARCHAR(20)	'<Name of your propagation range set in SAP S/4HANA>'	Propagation range for SAP S/4HANA Advanced Planning
`$G_PLAN_AREA`	VARCHAR(50)	'<Name of your planning area>'	SAP IBP planning area
`$G_ATP_CATEGORIES_STR`	VARCHAR(200)	'AG, AH, BF, SP, SO'	ATP categories for STRs maintained as a list
`$G_ATP_CATEGORIES_PROD`	VARCHAR(200)	'AA, AB, AC, AD, AI, AJ, AL'	ATP categories for production orders maintained as a list

Global Variable	Data Type	Default Value	Description
$G_ATP_CATEGORIES_PROC	VARCHAR(200)	'AG, AH, BF'	ATP categories for procurement receipts maintained as a list
$G_AGG_OBJ_TYPE_PROC	INTEGER	3	Aggregated object type for SAP S/4HANA Advanced Planning - procurement receipts
$G_AGG_OBJ_TYPE_STR	INTEGER	2	Aggregated object type for SAP S/4HANA Advanced Planning - STRs
$G_AGG_OBJ_TYPE_PROD	INTEGER	1	Aggregated object type for SAP S/4HANA Advanced Planning - production orders
$G_TIME_PROFILE_LEVEL	INTEGER	3	Time profile level - optional SAP IBP parameter used for disaggregation (Refer to the time profile in the planning area)
$G_ATP_CAT_NO_STR	INTEGER		Technical (helper) global variable used only in the script determining the number of ATP categories for stock transfer requisitions used at runtime
$G_ATP_CAT_NO_PROD	INTEGER		Technical (helper) global variable used only in the script determining the number of ATP categories for production orders used at runtime
$G_ATP_CAT_NO_PROC	INTEGER		Technical (helper) global variable used only in the script determining the number of ATP categories for procurement receipts used at runtime

Global Variable	Data Type	Default Value	Description
$G_ATP_HELPER	VARCHAR(10)		Technical (helper) global variable used only in the script
$G_PLANNING_VERSION	VARCHAR(10)	'000'	Planning version in SAP S/4HANA
$G_FILE_NAME	VARCHAR(100)	'IBPPPDS_TIME_PRO-FILE.csv'	File where the SAP IBP time profile selection is stored
$G_SCENARIO	VARCHAR(50)	''	SAP IBP planning scenario
$G_BATCH_COMMAND	VARCHAR(50)	'INSERT_UPDATE'	SAP IBP post-processing batch command (INSERT_UPDATE or DELETE)
$G_TIME_PROFILE	INTEGER	-1	SAP IBP time profile ID. By default, the time profile assigned to the planning area is used
$G_LOAD_DATE	DATETIME		Load date and time (calculated)
$G_LAST_STANDARD_LOAD_DATE	DATETIME		Last successful load date and time with standard settings (calculated)
$G_STANDARD_LOAD	VARCHAR(3)		Data load with standard settings (calculated)
$G_LAST_LOAD_DATE	DATETIME		Last successful load date and time (calculated)
$G_LOG_HELPER	VARCHAR(50)		Helper variable for printing to trace log

If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:

```
IF (<global variable> = '<default value>')
```

> **i Note**
>
> The propagation range defined in SAP S/4HANA acts as a filter. With that range, you define for which location materials and resources the data flow is relevant.

9. Map the $G_TIME_PROFILE_LEVEL global variable to the ID of the IBP time profile level for calendar week.
10. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▌ *More Actions* ❯ *Promote* ▌. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.17.2 IBP_PPDS_from_S4_Res_CapInfo

You can use the `IBP_PPDS_from_S4_Res_CapInfo` template as a starting point to transfer the resource capacity information for supply and usage per bucket from Advanced Planning (Production Planning and Detailed Scheduling (PP/DS)) in SAP S/4HANA to SAP Integrated Business Planning for Supply Chain (SAP IBP) time-series-based supply planning into key figures *Confirmed Capacity Supply* and *Confirmed Capacity Usage* on resource location level.

> **i Note**
>
> As a source table, the flat file created using the task `IBP_PPDS_to_file_IBP_TimeProfile` is used. Therefore, it is a prerequisite to run this data flow before executing the task `IBP_PPDS_from_S4_Res_CapInfo`.

The template consists of the following data flows:

Data Flow Name	Description	Source Function Module and Table	Target Key Figure
PPDS_IBP_CapaInfo	PP/DS capacity Information is read in PP/DS within an RFC Call, and the result is written into the key figures `CONFIRMEDCAPASUPPLY` and `CONFIRMEDCAPAUSAGE` in SAP IBP	`/SAPAPO/ SYPP_RES_CAPA_GET`	`CONFIRMEDCAPASUPPLY` `CONFIRMEDCAPAUSAGE`

Prerequisites for Creating a Task Based on the IBP_PPDS_from_S4_Res_CapInfo Template

Creating Datastores

1. If it is not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP IBP system as an instance, contact SAP.

2. Import the SAP IBP staging tables for key figures of the target planning area as follows:
   1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
   2. Expand the *Timeseries Folder* in the tree and select your planning area.
   3. Expand the planning area, select the staging table, and choose *Import*.

   > **i Note**
   >
   > The name of the staging table follows the pattern `SOPDD_STAGING_KFTAB_<Example Planning Area><Planning Area>` (do not use the table with the suffix `_REP`).
   >
   > Abbreviations may occur in the name of the calculation scenario.

3. If not yet available, create a datastore for your SAP S/4HANA system with the following data:

Name	A name of your choice, for example, **SAPS4HANA**
Description	A description of your choice, for example, **SAP S/4HANA Datastore**
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application Server	Your SAP S/4HANA application server

4. Load the definition of the `/SAPAPO/SYPP_RES_CAPA_GET` function module there as follows:
   1. Navigate to the *Tables* tab.
   2. Choose *Import Objects by Name*.
   3. Change the type to *Function*.

4. In the *Name* field, enter the name of the function module, and choose *OK*.
5. Check that the function module has been added to the list of tables.

> **i Note**
>
> If the datastore has several configurations, make sure the configuration of the system for loading the function module is the default one. If it isn't, make it the default and don't forget to save the changes.

How to Use This Template?

1. As source datastore, select the file format group datastore you created for the `IBP_PPDS_to_file_IBP_TimeProfile` task.
2. As target datastore, select the datastore that represents your SAP IBP system.
3. Open the data flow for editing by choosing ▶ *Actions* ❯ *Edit* ◀.

> **i Note**
>
> In the note boxes of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

4. Delete the `CallRFC` transform in the data flow, replace it with a transform of type *Web Service or Function Call*, and name it `CallRFC` again.
5. Create the mapping arrows from `PrepareRFC` to `CallRFC` and from `CallRFC` to `HandleOutput`.
6. Choose the `CallRFC` transform. Choose *Select Function: Web Service or RFC Function* on the output node `CallRFC`. Select the function module you have uploaded from your SAP S/4HANA datastore, and map all first-level nodes of `PrepareRFC` (for example, `IT_DATETIME_INTERVAL`) to the corresponding nodes of `CallRFC`.
7. Under *Execution Properties*, you can modify the default values of the following global variables:

Global Variable	Data Type	Default Value	Description
`$G_TIME_PROFILE_LEVEL`	INTEGER	3	Time profile level - optional SAP IBP parameter used for disaggregation (Refer to the time profile in the planning area)
`$G_PROPAGATION_RANGE`	VARCHAR(20)	'<Name of your propagation range set in SAP S/4HANA>'	Propagation range for SAP S/4HANA Advanced Planning
`$G_PLAN_AREA`	VARCHAR(50)	'<Name of your planning area>'	SAP IBP planning area

Global Variable	Data Type	Default Value	Description
$G_TIME_PROFILE	INTEGER	-1	SAP IBP time profile ID. By default, the time profile assigned to the planning area is used
$G_PLANNING_VERSION	VARCHAR(10)	'000'	Planning version in SAP S/4HANA
$G_FILE_NAME	VARCHAR(100)	'IBPPPDS_TIME_PRO-FILE.csv'	File where the SAP IBP time profile selection is stored
$G_SCENARIO	VARCHAR(50)	''	SAP IBP planning scenario
$G_BATCH_COMMAND	VARCHAR(50)	'INSERT_UPDATE'	SAP IBP post-processing batch command (INSERT_UPDATE or DELETE)
$G_LOAD_DATE	DATETIME		Load date and time (calculated)
$G_LAST_STANDARD_LOAD_DATE	DATETIME		Last successful load date and time with standard settings (calculated)
$G_STANDARD_LOAD	VARCHAR(3)		Data load with standard settings (calculated)
$G_LAST_LOAD_DATE	DATETIME		Last successful load date and time (calculated)
$G_LOG_HELPER	VARCHAR(50)		Helper variable for printing to trace log

If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:

```
IF (<global variable> = '<default value>')
```

> **i Note**
>
> The propagation range defined in SAP S/4HANA acts as a filter. With that range, you define for which location materials and resources the data flow is relevant.

8. Map the $G_TIME_PROFILE_LEVEL global variable to the ID of the time profile level in SAP IBP for calendar week.
9. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▌ *More Actions* ❯ *Promote* ▐. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.17.3 IBP_PPDS_to_file_IBP_TimeProfile

You can use the `IBP_PPDS_to_file_IBP_TimeProfile` template as a starting point to write time intervals of a definable period to a CSV file stored in SAP Cloud Integration for data services. These time intervals are used in subsequest data flows to restrict the time selection when transferring data between Advanced Planning (Production Planning and Detailed Scheduling (PP/DS)) in SAP S/4HANA and SAP Integrated Business Planning for Supply Chain (SAP IBP) time-series-based supply planning.

> **i Note**
>
> It is important to run this data flow at least weekly or, even better, each time data extraction from PP/DS to SAP IBP is required to get to the latest definitions of the time profile. The time profile in this template is stored at calendar week level (PERIODID4). This requires adjustment if the time profile in the implementation project is different.

The template consists of the following data flows:

Data Flow Name	Description	Source Key Figure and Master Data Fields	Target Function Module and Table
IBPPPDS_TimeProfile	Extract SAP IBP time profile to file on calendar week level	PERIODID PERIODID4 TSTFR TSTTO WEEKWEIGTH	CSV Flat file

Prerequisites for Creating a Task Based on the IBP_PPDS_to_file_IBP_TimeProfile Template

Creating Datastores

1. If it is not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP IBP system as an instance, contact SAP.

2. Import the time period that you want to use in subsequent data flows to define the periods to read data from.
   1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
   2. Expand the *Calendar Folder* in the tree and select your planning area.
   3. Expand the planning area, select `SOPDD_TIMEPERIOD_<Name of the time profile>`, and choose *Import*.

   > **i Note**
   >
   > The name of the time period follows the pattern `SOPDD_TIMEPERIOD_<Name of the time profile>`. Do not import the staging table `SOPDD_STAG_TIMEPERIOD_<Name of the time profile>`.
   >
   > Abbreviations may occur in the name of the time profile.

3. If not yet available, create a datastore of the type *File Format Group*.
   Prerequisite: The file format type datastore requires a root directory on SAP Data Services Agent. You have to create the root directory, publish it on the agent, and add it to the configuration of the agent before you create the following datastore.
   Provide the following data for the new datastore:

Name	A name of your choice, for example, **FILES**
Description	A description of your choice, for example, **File share**
Type	File Format Group
Agent	The SAP Data Services Agent in your landscape

Root Directory	Name of the directory on the agent system for storing the output files, for example, **C:\HCI_Share** in the case of a Microsoft Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. Create the file format as follows:
   1. Navigate to the *File Formats* tab.
   2. Choose *Create File Format*.
   3. Choose *Create From Scratch*.
      Provide the following data for the new datastore:

Data Type	Value
Name	A name of your choice, for example, **IBPPPDS_TIME_PROFILE**
Description	A description of your choice
First row contains column names	X

   For all other fields, use the default values.

   4. Select your newly created file format and navigate to *Columns*. Choose *Add Column* and provide the following data:

Data Type	Value
PERIODID4	integer
START_TIMED	varchar(14)
END_TIME	varchar(14)

How to Use This Template?

1. As source datastore, select the datastore that represents your SAP IBP system.
2. As target datastore, select the datastore of type *File Format Group* that you have previously created. Choose *Save and Define Data Flow*.
3. Open the data flow for editing by choosing ▌▶ *Actions* ❯ *Edit* ▐.

> **i** Note
>
> In the note boxes of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

4. Replace the source table with your own time period table and recreate the mapping arrow from the source table to the `QueryAndFilter` transform.
5. Under *Execution Properties*, you can modify the default values of the following global variables:

Global Variable	Data Type	Default Value	Description
`$G_FILE_NAME`	VARCHAR(100)	'IBPPPDS_TIME_PRO-FILE.csv'	Target file for time profile extraction from SAP IBP
`$G_FIRST_SELECTION_WEEK`	INTEGER	1	First selection week (before or after the current week)
`$G_LAST_SELECTION_WEEK`	INTEGER	20	Last selection week (before or after the current week)
`$G_LOG_HELPER`	VARCHAR(50)	''	Helper field for writing log entries in the preload script

If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:

```
IF (<global variable> = '<default value>')
```

6. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ▶ *Promote* ▶. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.17.4 IBP_PPDS_to_S4_Distribution_BucketOrders

You can use the `IBP_PPDS_to_S4_Distribution_BucketOrders` template as a starting point to transfer weekly transport receipts quantity from SAP Integrated Business Planning for Supply Chain (SAP IBP) time-series-based supply planning to Advanced Planning (Production Planning and Detailed Scheduling (PP/DS)) in SAP S/4HANA as stock transfer requisitions (STRs).

> **i Note**
>
> If you are also running the task `IBP_PPDS_to_S4_Production_BucketOrders`, we recommend to group both data flows together into a process flow so that they are executed together.

The template consists of the following data flows:

Data Flow Name	Description	Source Key Figure and Master Data Fields	Target Function Module and Table
IBP_to_PPDS_Distribution_BucketOrder	Sends distribution quantities (key figure TRANSPORT) from SAP IBP as bucket orders and transfers them to PP/DS using an RFC call	PRDID LOCID LOCFR PERIODID4 TSTFR TSTTO TRANSPORT UOMID	/SAPAPO/ SYPP_CREATE_BUCSTR

Time Aggregation

This template uses calendar weeks for time aggregation, as the field PERIODID4 (level 3, calendar weeks) is mapped in the data flow. Please note that it is not sufficient to map it in the PERIODIDn field in the QueryAndFilter transform without doing any follow-on processing with the QueryAndFilter.PERIODIDn field. For this reason, there is the following included in the Mapping transform for the DATE_FROM and DATE_TO fields:

```
ifthenelse(QueryAndFilter.PERIODIDn >0, QueryAndFilter.TSTFR, QueryAndFilter.TSTFR)
```

This way, it is made sure that SAP Cloud Integration for data services requests the PERIODID4 field from SAP IBP, which is used to calculate the DATE_FROM and DATE_TO fields.

Prerequisites for Creating a Task Based on the IBP_PPDS_to_S4_Distribution_BucketOrders Template

Creating Datastores

1. If it is not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning

Data Type	Value
Instance	Your SAP IBP system

> **i Note**
>
> If you cannot select your SAP IBP system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading from the planning area in SAP IBP as follows:
   1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
   2. Expand the *Calculation Scenario Folder* in the tree and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

   > **i Note**
   >
   > The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`.
   >
   > Abbreviations may occur in the name of the calculation scenario.

3. If not yet available, create a datastore of the type *File Format Group*. This file is created because every data flow in SAP Cloud Integration for data services needs to have a target, and the function module cannot be a target. SAP IBP cannot be the target either, as it is already the source, and it cannot be both a source and a target. Therefore, the file is always created, but it does not contain any useful information.
   Prerequisite: The file format type datastore requires a root directory on SAP Data Services Agent. You have to create the root directory, publish it on the agent, and add it to the configuration of the agent before you create the following datastore.
   Provide the following data for the new datastore:

Name	A name of your choice, for example, **FILES**
Description	A description of your choice, for example, **File share**
Type	File Format Group
Agent	The SAP Data Services Agent in your landscape
Root Directory	Name of the directory on the agent system for storing the output files, for example, **C:\HCI_Share** in the case of a Microsoft Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. On the *File Formats* tab, choose ▶ *Create File Format* ▶ *Create from Scratch* ▶.
   Name the new file format **DUMMY_OUPUT_FILE**. If you choose a different name, you need to copy the data flows to a new target later.
   Enter **Dummy output file with one dummy field** as a description, and choose *OK*.
   Choose *Add Column* on the *Columns* tab. Enter **Dummy** in the *Name* field, and **Dummy field** in the *Description* field. If you choose a different name, you need to adapt the mapping in the `Target_Query` transform of the `IBP_PPDS_to_S4_Distribution_BucketOrders` data flow accordingly.

Set the data type to VARCHAR and the length to **1**, then choose *Submit*.

5. If not yet available, create a datastore for your SAP S/4HANA system with the following data:

Name	A name of your choice, for example, **SAPS4HANA**
Description	A description of your choice, for example, `SAP S/4HANA Datastore`
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application Server	Your SAP S/4HANA application server

6. Load the definition of the `/SAPAPO/SYPP_CREATE_BUCSTR` function module there as follows:
   1. Navigate to the *Tables* tab.
   2. Choose *Import Objects by Name*.
   3. Change the type to *Function*.
   4. In the *Name* field, enter the name of the function module, and choose *OK*.
   5. Check that the function module has been added to the list of tables.

> **i Note**
>
> If the datastore has several configurations, make sure the configuration of the system for loading the function module is the default one. If it isn't, make it the default and don't forget to save the changes.

Defining the Planning Unit of Measure

To be able to run the `IBP_PPDS_to_S4_Distribution_BucketOrders`, a planning unit of measure (UoM) is required in SAP IBP. The conversion factor 1:1 needs to be maintained between this planning unit of measure and the base unit of measure for all products in SAP IBP. By default, this planning UoM is called **1BU** in the SAP Cloud Integration for data services template.

To make sure that your planning UoM is adapted to the template, follow the instructions that apply to your case:

- If there is a planning UoM already available in SAP IBP called **1BU**, no further steps are required.
- If there is a planning UoM already available in SAP IBP, but it is called differently, change the value of the `$G_AGGREGATION_UOM` global variable in your SAP Cloud Integration for data services task to the name of your own planning UoM.
- If there is no planning UoM available in SAP IBP yet, create one in one of the following ways:
  - Create it manually and maintain the conversion factor between this planning UoM and the base UoM as 1:1 in SAP IBP.
  - Create it using the `IBP_MD_UOM_w_Text_AddOn` and `IBP_MD_UOMConversionFactor_AddOn` data flows of the `IBP_MD_S4_ERP_AddOn` template and create the conversion factor 1:1 between this planning UoM and the base UoM using the SAP Cloud Integration for data services task.
  If the `$G_UNIFIED_BASE_UOM` global parameter is not set to an empty string, the data flows add both a unified base unit of measure to SAP IBP with the name `$G_UNIFIED_BASE_UOM` and the necessary 1:1 mappings.

Defining the Material Type Filter

Check the global variable $G_ MATTYPEID_FG and ensure it is mapped with the value identifying the finished goods in the attribute *Material Type ID* of the master data type *Product* in your SAP IBP system.

How to Use This Template?

1. As source datastore, select the datastore that represents your SAP IBP system.
2. As target datastore, select the datastore of type *File Format Group* that you have previously created. Choose *Save and Define Data Flow*.
   In case you have named the file format other than **DUMMY_OUTPUT_FILE**, select the data flow, and choose *Copy to new target*. Enter a name for the data flow, select the target file format, and choose *Copy Data Flow*. Then, delete the original data flow and the target table DUMMY_OUTPUT_FILE.
3. Open the data flow for editing by choosing ▷ *Actions* ▷ *Edit* ▷.

   > **i Note**
   >
   > In the note boxes of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

4. Replace the source table with your own calculation view and recreate the mapping arrow from the source table to the QueryAndFilter transform.
5. Delete the CallRFC transform in all the data flows, replace it with a transform of type *Web Service or Function Call*, and name it CallRFC again.
6. Create the mapping arrows from PrepareRFC to CallRFC and from CallRFC to GetRFCOutput.
7. Choose the CallRFC transform. Choose *Select Function: Web Service or RFC Function* on the output node CallRFC. Select the function module you have uploaded from your SAP S/4HANA datastore, and map all first-level nodes of PrepareRFC (for example, IT_BUCKET) to the corresponding nodes of CallRFC.
8. Under *Execution Properties*, you can modify the default values of the following global variables:

Global Variable	Data Type	Default Value	Description
$G_PLANNING_VERSION	VARCHAR(10)	'000'	Planning version in SAP S/4HANA
$G_PLAN_AREA	VARCHAR(50)	'<Name of your planning area>'	SAP IBP planning area

Global Variable	Data Type	Default Value	Description
$G_IBP_USER_ID	VARCHAR(50)		Optional parameter enabling filtering data using permission filters and attribute permissions when triggering the task from SAP IBP. Not mapped when triggering the task from SAP Cloud Integration for data services.
$G_IBP_FILTER_ID	VARCHAR(100)		Optional parameter enabling filtering data using permission filters and attribute permissions when triggering the task from SAP IBP. Not mapped when triggering the task from SAP Cloud Integration for data services.
$G_AGGREGATION_UOM	VARCHAR(3)	'1BU'	Unit of measure for aggregation of key figures in SAP IBP
$G_FIRST_SELECTION_WEEK	INTEGER	1	First selection week (before or after the current week)
$G_LAST_SELECTION_WEEK	INTEGER	20	Last selection week (before or after the current week)
$G_LINEITEMS_PER_PACKAGE	INTEGER	10000	Approximate line items per package
$G_JOB_GUID	VARCHAR(32)	gen_uuid()	
$G_LOG_HELPER	VARCHAR(50)	''	Helper variable for printing to trace log
$G_MATTYPEID_FG	VARCHAR(4)	'FERT'	Material type filter for extracting key figures only for SAP IBP products marked as finished goods

If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:

```
IF (<global variable> = '<default value>')
```

9. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▌ *More Actions* ❯ *Promote* ▐. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.17.5 IBP_PPDS_to_S4_Forecasts

You can use the `IBP_PPDS_to_S4_Forecasts` template as a starting point to transfer weekly consensus demand quantity from SAP Integrated Business Planning for Supply Chain (SAP IBP) time-series-based supply planning to Advanced Planning (Production Planning and Detailed Scheduling (PP/DS)) in SAP S/4HANA as planned independent requirements.

> i Note
>
> This key figure is used as an example in the SAP Cloud Integration for data services template. In case you have, for example, independent demand at the same location, you need to adjust the default mapping.

The template consists of the following data flows:

Data Flow Name	Description	Source Key Figure and Master Data Fields	Target Function Module and Table
`IBP_to_PPDS_Forecast`	Reads the key figure *Consensus Demand Plan Qty* (CONSENSUSDEMANDQTY) from SAP IBP planning area and transfers to PP/DS using an RFC call	`PRDID` `LOCID` `PERIODID4` `TSTFR` `TSTTO` `CONSENSUSDEMANDQTY`	`/SAPAPO/SYPP_CREATE_FC`

Time Aggregation

This template uses calendar weeks for time aggregation, as the field `PERIODID4` (level 3, calendar weeks) is mapped in the data flow. Please note that it is not sufficient to map it in the `PERIODID4` field in the `QueryAndFilter` transform without doing any follow-on processing with the `QueryAndFilter.PERIODIDn` field. For this reason, there is the following included in the `Mapping` transform for the `DATE_FROM` and `DATE_TO` fields:

```
ifthenelse(QueryAndFilter.PERIODIDn >0, QueryAndFilter.TSTFR, QueryAndFilter.TSTFR)
```

This way, it is ensured that SAP Cloud Integration for data services requests the `PERIODID4` field from SAP IBP, which is used to calculate the `DATE_FROM` and `DATE_TO` fields.

Prerequisites for Creating a Task Based on the IBP_PPDS_to_S4_Forecasts Template

Creating Datastores

1. If it is not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore
Type	Integrated Business Planning
Instance	Your SAP IBP system

> i Note
>
> If you cannot select your SAP IBP system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading from the planning area in SAP IBP as follows:
    1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
    2. Expand the *Calculation Scenario Folder* in the tree and select your planning area.
    3. Expand the planning area, select the calculation scenario, and choose *Import*.

    > i Note
    >
    > The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`.
    >
    > Abbreviations may occur in the name of the calculation scenario.

3. If not yet available, create a datastore of the type *File Format Group*. This file is created because every data flow in SAP Cloud Integration for data services needs to have a target, and the function module cannot be a target. SAP IBP cannot be the target either, as it is already the source, and it cannot be both a source and a target. Therefore, the file is always created, but it does not contain any useful information.
   Prerequisite: The file format type datastore requires a root directory on SAP Data Services Agent. You have to create the root directory, publish it on the agent, and add it to the configuration of the agent before you create the following datastore.
   Provide the following data for the new datastore:

Name	A name of your choice, for example, **FILES**

Description	A description of your choice, for example, **File share**
Type	File Format Group
Agent	The SAP Data Services Agent in your landscape
Root Directory	Name of the directory on the agent system for storing the output files, for example, **C:\HCI_Share** in the case of a Microsoft Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. On the *File Formats* tab, choose ▶ *Create File Format* ❯ *Create from Scratch* ❮.
   Name the new file format **DUMMY_OUPUT_FILE**. If you choose a different name, you need to copy the data flows to a new target later.
   Enter **Dummy output file with one dummy field** as a description, and choose *OK*.
   Choose *Add Column* on the *Columns* tab. Enter **Dummy** in the *Name* field, and **Dummy field** in the *Description* field. If you choose a different name, you need to adapt the mapping in the `Target_Query` transform of the `IBP_PPDS_to_S4_Forecasts` data flow accordingly.
   Set the data type to VARCHAR and the length to **1**, then choose *Submit*.

5. If not yet available, create a datastore for your SAP S/4HANA system with the following data:

Name	A name of your choice, for example, **SAPS4HANA**
Description	A description of your choice, for example, **SAP S/4HANA Datastore**
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application Server	Your SAP S/4HANA application server

6. Load the definition of the `/SAPAPO/SYPP_CREATE_FC` function module there as follows:
   1. Navigate to the *Tables* tab.
   2. Choose *Import Objects by Name*.
   3. Change the type to *Function*.
   4. In the *Name* field, enter the name of the function module, and choose *OK*.
   5. Check that the function module has been added to the list of tables.

> **i Note**
>
> If the datastore has several configurations, make sure the configuration of the system for loading the function module is the default one. If it isn't, make it the default and don't forget to save the changes.

Defining the Planning Unit of Measure

To be able to run the `IBP_PPDS_to_S4_Forecasts`, a planning unit of measure (UoM) is required in SAP IBP. The conversion factor 1:1 needs to be maintained between this planning unit of measure and the base unit of

measure for all products in SAP IBP. By default, this planning UoM is called **1BU** in the SAP Cloud Integration for data services template.

To make sure that your planning UoM is adapted to the template, follow the instructions that apply to your case:

- If there is a planning UoM already available in SAP IBP called **1BU**, no further steps are required.
- If there is a planning UoM already available in SAP IBP, but it is called differently, change the value of the `$G_AGGREGATION_UOM` global variable in your SAP Cloud Integration for data services task to the name of your own planning UoM.
- If there is no planning UoM available in SAP IBP yet, create one in one of the following ways:
  - Create it manually and maintain the conversion factor between this planning UoM and the base UoM as 1:1 in SAP IBP.
  - Create it using the `IBP_MD_UOM_w_Text_AddOn` and `IBP_MD_UOMConversionFactor_AddOn` data flows of the `IBP_MD_S4_ERP_AddOn` template and create the conversion factor 1:1 between this planning UoM and the base UoM using the SAP Cloud Integration for data services task.
    If the `$G_UNIFIED_BASE_UOM` global parameter is not set to an empty string, the data flows add both a unified base unit of measure to SAP IBP with the name `$G_UNIFIED_BASE_UOM` and the necessary 1:1 mappings.

How to Use This Template?

1. As source datastore, select the datastore that represents your SAP IBP system.
2. As target datastore, select the datastore of type *File Format Group* that you have previously created. Choose *Save and Define Data Flow*.
   In case you have named the file format other than **DUMMY_OUTPUT_FILE**, select the data flow, and choose *Copy to new target*. Enter a name for the data flow, select the target file format, and choose *Copy Data Flow*. Then, delete the original data flow and the target table DUMMY_OUTPUT_FILE.
3. Open the data flow for editing by choosing ▍▶ *Actions* ❭ *Edit* ▍.

   > **i Note**
   >
   > In the note boxes of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

4. Replace the source table with your own calculation view and recreate the mapping arrow from the source table to the `QueryAndFilter` transform.
5. Delete the `CallRFC` transform in all the data flows, replace it with a transform of type *Web Service or Function Call*, and name it `CallRFC` again.
6. Create the mapping arrows from `PrepareRFC` to `CallRFC` and from `CallRFC` to `GetRFCOutput`.
7. Choose the `CallRFC` transform. Choose *Select Function: Web Service or RFC Function* on the output node `CallRFC`. Select the function module you have uploaded from your SAP S/4HANA datastore, and map all first-level nodes of `PrepareRFC` (for example, `REQUIREMENTS`) to the corresponding nodes of `CallRFC`.

8. Under *Execution Properties*, you can modify the default values of the following global variables:

Global Variable	Data Type	Default Value	Description
$G_PLANNING_VERSION	VARCHAR(10)	'000'	Planning version in SAP S/4HANA
$G_PLAN_AREA	VARCHAR(15)	'<Name of your planning area>'	SAP IBP planning area
$G_LOGICAL_SYSTEM	VARCHAR(12)	<Name of logical system>	Logical system of target RFC
$G_IBP_USER_ID	VARCHAR(50)		Optional parameter enabling filtering data using permission filters and attribute permissions when triggering the task from SAP IBP. Not mapped when triggering the task from SAP Cloud Integration for data services.
$G_IBP_FILTER_ID	VARCHAR(100)		Optional parameter enabling filtering data using permission filters and attribute permissions when triggering the task from SAP IBP. Not mapped when triggering the task from SAP Cloud Integration for data services.
$G_AGGREGATION_UOM	VARCHAR(3)	'1BU'	Unit of measure for aggregation of key figures in SAP IBP
$G_FIRST_SELECTION_WEEK	INTEGER	1	First selection week (before or after the current week)
$G_LAST_SELECTION_WEEK	INTEGER	20	Last selection week (before or after the current week)
$G_LINEITEMS_PER_PACKAGE	INTEGER	10000	Approximate line items per package
$G_JOB_GUID	VARCHAR(32)	gen_uuid()	

Global Variable	Data Type	Default Value	Description
$G_LOG_HELPER	VARCHAR(50)	''	Helper variable for printing to trace log

If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:

```
IF (<global variable> = '<default value>')
```

9. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▶ *More Actions* ❯ *Promote* ❯. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.4.17.6 IBP_PPDS_to_S4_Production_BucketOrders

You can use the BP_PPDS_to_S4_Production_BucketOrders template as a starting point to transfer weekly net new planned production receipts and production receipts from SAP Integrated Business Planning for Supply Chain (SAP IBP) time-series-based supply planning to Advanced Planning (Production Planning and Detailed Scheduling (PP/DS)) in SAP S/4HANA as time-series-based bucket orders.

> i Note
>
> If you are also running the task IBP_PPDS_to_S4_Distribution_BucketOrders, we recommend to group both data flows together into a process flow so that they are executed together.

The template consists of the following data flows:

Data Flow Name	Description	Source Key Figure and Master Data Fields	Target Function Module and Table
IBP_to_PPDS_ProdTgtSupply	Sends production quantities (key figure NETNEWPLANNEDPRODREC) from SAP IBP as bucket orders and target supply (key figure PRODUCTION) and transfers them to PP/DS using an RFC call	PRDID LOCID SOURCEID PERIODID4 TSTFR TSTTO NETNEWPLANNEDPRODREC PRODUCTION UOMID	/SAPAPO/ SYPP_CREATE_BUCORD

Time Aggregation

This template uses calendar weeks for time aggregation, as the field PERIODID4 (level 3, calendar weeks) is mapped in the data flow. Please note that it is not sufficient to map it in the PERIODIDn field in the QueryAndFilter transform without doing any follow-on processing with the QueryAndFilter.PERIODIDn field. For this reason, there is the following included in the Mapping transform for the DATE_FROM and DATE_TO fields:

```
ifthenelse(QueryAndFilter.PERIODIDn >0, QueryAndFilter.TSTFR, QueryAndFilter.TSTFR)
```

This way, it is made sure that SAP Cloud Integration for data services requests the PERIODID4 field from SAP IBP, which is used to calculate the DATE_FROM and DATE_TO fields.

Prerequisites for Creating a Task Based on the IBP_PPDS_to_S4_Production_BucketOrders Template

Creating Datastores

1. If it is not yet available, create a datastore that represents your SAP IBP system. Provide the following data for the new datastore:

Data Type	Value
Name and Description	Name and description of the datastore

Data Type	Value
Type	Integrated Business Planning
Instance	Your SAP IBP system

> **i** Note
>
> If you cannot select your SAP IBP system as an instance, contact SAP.

2. Import the calculation scenario to be used for reading dependent demand quantity from the planning area in SAP IBP as follows:
   1. Select the datastore that represents your SAP IBP system, and choose *Import Objects*.
   2. Expand the *Calculation Scenario Folder* in the tree and select your planning area.
   3. Expand the planning area, select the calculation scenario, and choose *Import*.

   > **i** Note
   >
   > The name of the calculation scenario follows the pattern `SAPSOPG.SOP<Planning Area Name><Planning Area Version><Planning Area Name>`.
   >
   > Abbreviations may occur in the name of the calculation scenario.

3. If not yet available, create a datastore of the type *File Format Group*. This file is created because every data flow in SAP Cloud Integration for data services needs to have a target, and the function module cannot be a target. SAP IBP cannot be the target either, as it is already the source, and it cannot be both a source and a target. Therefore, the file is always created, but it does not contain any useful information.
   Prerequisite: The file format type datastore requires a root directory on SAP Data Services Agent. You have to create the root directory, publish it on the agent, and add it to the configuration of the agent before you create the following datastore.
   Provide the following data for the new datastore:

Name	A name of your choice, for example, **FILES**
Description	A description of your choice, for example, **File share**
Type	File Format Group
Agent	The SAP Data Services Agent in your landscape
Root Directory	Name of the directory on the agent system for storing the output files, for example, **C:\HCI_Share** in the case of a Microsoft Windows agent. The directory also needs to be maintained in the configuration of the agent as an accessible directory.

4. On the *File Formats* tab, choose ▌▶ *Create File Format* ❯ *Create from Scratch* ❯ .
   Name the new file format **DUMMY_OUPUT_FILE**. If you choose a different name, you need to copy the data flows to a new target later.
   Enter **Dummy output file with one dummy field** as a description, and choose *OK*.

Choose *Add Column* on the *Columns* tab. Enter **Dummy** in the *Name* field, and **Dummy field** in the *Description* field. If you choose a different name, you need to adapt the mapping in the `Target_Query` transform of the `IBP_PPDS_to_S4_Production_BucketOrders` data flow accordingly. Set the data type to `VARCHAR` and the length to **1**, then choose *Submit*.

5. If not yet available, create a datastore for your SAP S/4HANA system with the following data:

Name	A name of your choice, for example, **SAPS4HANA**
Description	A description of your choice, for example, **SAP S/4HANA Datastore**
Type	SAP Business Suite Applications
Agent	The SAP Data Services Agent in your landscape
Application Server	Your SAP S/4HANA application server

6. Load the definition of the `/SAPAPO/SYPP_CREATE_BUCORD` function module there as follows:
   1. Navigate to the *Tables* tab.
   2. Choose *Import Objects by Name*.
   3. Change the type to *Function*.
   4. In the *Name* field, enter the name of the function module, and choose *OK*.
   5. Check that the function module has been added to the list of tables.

> **i Note**
>
> If the datastore has several configurations, make sure the configuration of the system for loading the function module is the default one. If it isn't, make it the default and don't forget to save the changes.

Defining the Planning Unit of Measure

To be able to run the `IBP_PPDS_to_S4_Production_BucketOrders`, a planning unit of measure (UoM) is required in SAP IBP. The conversion factor 1:1 needs to be maintained between this planning unit of measure and the base unit of measure for all products in SAP IBP. By default, this planning UoM is called 1BU in the SAP Cloud Integration for data services template.

To make sure that your planning UoM is adapted to the template, follow the instructions that apply to your case:

- If there is a planning UoM already available in SAP IBP called **1BU**, no further steps are required.
- If there is a planning UoM already available in SAP IBP, but it is called differently, change the value of the `$G_AGGREGATION_UOM` global variable in your SAP Cloud Integration for data services task to the name of your own planning UoM.
- If there is no planning UoM available in SAP IBP yet, create one in one of the following ways:
  - Create it manually and maintain the conversion factor between this planning UoM and the base UoM as 1:1 in SAP IBP.
  - Create it using the `IBP_MD_UOM_w_Text_AddOn` and `IBP_MD_UOMConversionFactor_AddOn` data flows of the `IBP_MD_S4_ERP_AddOn` template and create the conversion factor 1:1 between this planning UoM and the base UoM using the SAP Cloud Integration for data services task.

If the `$G_UNIFIED_BASE_UOM` global parameter is not set to an empty string, the data flows add both a unified base unit of measure to SAP IBP with the name `$G_UNIFIED_BASE_UOM` and the necessary 1:1 mappings.

Defining the Location Type Filter

Check the global variable `$G_LOCTYPE_PLANT` and ensure it is mapped with the value identifying the plants (factories) in the attribute *Location Type* of the master data type *Location* in your SAP IBP system.

How to Use This Template?

1. As source datastore, select the datastore that represents your SAP IBP system.
2. As target datastore, select the datastore of type *File Format Group* that you have previously created. Choose *Save and Define Data Flow*.
   In case you have named the file format other than **DUMMY_OUTPUT_FILE**, select the data flow, and choose *Copy to new target*. Enter a name for the data flow, select the target file format, and choose *Copy Data Flow*. Then, delete the original data flow and the target table DUMMY_OUTPUT_FILE.
3. Open the data flow for editing by choosing ▮ *Actions* ❯ *Edit* ▮.

   > **i Note**
   >
   > In the note boxes of the data flow, you can find instructions about adjusting the data flow to fit your needs. This document provides you with more detailed information about the required adjustments.

4. Replace the source table with your own calculation view and recreate the mapping arrow from the source table to the `QueryAndFilter` transform.
5. Delete the `CallRFC` transform in all the data flows, replace it with a transform of type *Web Service or Function Call*, and name it `CallRFC` again.
6. Create the mapping arrows from `PrepareRFC` to `CallRFC` and from `CallRFC` to `GetRFCOutput`.
7. Choose the `CallRFC` transform. Choose *Select Function: Web Service or RFC Function* on the output node `CallRFC`. Select the function module you have uploaded from your SAP S/4HANA datastore, and map all first-level nodes of `PrepareRFC` (for example, `IT_BUCKET`) to the corresponding nodes of `CallRFC`.
8. Under *Execution Properties*, you can modify the default values of the following global variables:

Global Variable	Data Type	Default Value	Description
`$G_PLAN_AREA`	VARCHAR(15)	'<Name of your planning area>'	SAP IBP planning area
`$G_PLANNING_VERSION`	VARCHAR(3)	'000'	Planning version in SAP S/4HANA
`$G_AGGREGATION_UOM`	VARCHAR(3)	'1BU'	Unit of measure for aggregation of key figures in SAP IBP

Global Variable	Data Type	Default Value	Description
$G_FIRST_SELECTION_WEEK	INTEGER	1	First selection week (before or after the current week)
$G_LAST_SELECTION_WEEK	INTEGER	20	Last selection week (before or after the current week)
$G_IBP_USER_ID	VARCHAR(50)		Optional parameter enabling filtering data using permission filters and attribute permissions when triggering the task from SAP IBP. Not mapped when triggering the task from SAP Cloud Integration for data services.
$G_IBP_FILTER_ID	VARCHAR(100)		Optional parameter enabling filtering data using permission filters and attribute permissions when triggering the task from SAP IBP. Not mapped when triggering the task from SAP Cloud Integration for data services.
$G_LINEITEMS_PER_PACKAGE	INTEGER	10000	Approximate line items per package
$G_JOB_GUID	VARCHAR(32)	gen_uuid()	
$G_LOG_HELPER	VARCHAR(50)	''	Helper variable for printing to trace log
$G_LOCTYPE_PLANT	VARCHAR(20)	'PLANT'	Location type filter for extracting key figures only for SAP IBP locations marked as plant

If you change the default value of any of the global variables above, also update the corresponding row in the preload script accordingly as follows:

```
IF (<global variable> = '<default value>')
```

9. Choose *Validate* to check your task. If required, make the necessary corrections.

Executing the Task

To execute a task, go to the *Projects* view, and choose *Run Now*. You can check the task run (view a log of errors and the number of records transferred) by selecting your task, and choosing *View History*.

After a thorough testing of your task, you may decide to schedule it for a periodic run. To do this, first promote your task to the production environment by selecting the task, and choosing ▌ *More Actions* ❯ *Promote* ▌. Then switch to the productive view. There, select your task, and choose *Schedule*, then *New*, and enter the required data.

7.5   General Purpose Templates

The general purpose templates contain information required by SAP Integrated Business Planning for Supply Chain to process the data after it is loaded. The templates contain global variables, preload scripts, and postload scripts.

7.5.1 SOP_APO_Task

This template has APO as source datastore, and SOP as target datastore. The template includes the preload script and postload script, as well as default values for the global variables.

You can use this task to define your own data flows from scratch.

7.5.2 SOP_ECC_Task

This template has ECC as source datastore, and SOP as target datastore. The template includes the preload script and postload script, as well as default values for the global variables.

You can use this task to define your own data flows from scratch.

7.5.3 SOP_File_Task

This template has a file format group as source datastore, and SOP as target datastore. The template includes the preload script and postload script, as well as default values for the global variables.

You can use this task to define your own data flows from scratch.

8 Global Variables

Global variables are symbolic placeholders. When a task runs, these placeholders are populated with values. This allows users flexibility of run-time values used in extractions.

> **i Note**
>
> Certain global variables are used by the SAP Integrated Business Planning for Supply Chain applications to process the data after it is loaded. If these global variables are not included in the task, then data will be loaded, but SAP Integrated Business Planning for Supply Chain cannot process it as expected. The required global variables are: `$G_PLAN_AREA`, `$G_SCENARIO`, `$G_TIME_PROFILE`, and `$G_BATCH_COMMAND`.

8.1 Setting Global Variables

Depending on your requirements and environment, allow the default values or set values in one of the following locations:

Option	Description
Run Now dialog box	Use for testing.
	From the *Projects* tab, select a task. From the *Actions* menu, choose *Run Now*.
Global Variables under *Execution Properties* tab of a task	Use for loading data.
	From the *Projects* tab, select a task. From the *Actions* menu, choose *Edit*. Click the *Execution Properties* tab.
Preload script	Use for testing and loading data.
If you do not specify values for global variables, during task execution the preload script populates the global variables with default values.	From the *Projects* tab, select a task. From the *Actions* menu, choose *Edit*. Click the *Execution Properties* tab, then the *Preload* tab under *Scripts*.

8.2 Shared Global Variables

Shared global variables are included in all SAP Integrated Business Planning for Supply Chain data integration tasks. Most tasks contain additional unique global variables.

Global variables contain default values which are used if the value has not been changed. As needed, change the value of each global variable per your requirements and environment. Certain global variables are used by the SAP Integrated Business Planning for Supply Chain applications to process the data after it is loaded. The default value and information about each of the shared global variables is shown in the following table:

Shared Global Variables

Required by SAP Integrated Business Planning for Supply Chain	Global Variable	Description	Data Type	Default Value for Master Data Task	Default Value for Key Figure Data Task
Yes	`$G_PLAN_AREA`	The planning are value is defined in SAP Integrated Business Planning for Supply Chain, in the *Configuration* app. The value must contain only uppercase alphanumeric characters and begin with a letter. Spaces are not allowed.	varchar(50)	"	`'SAPMODEL1'`
Yes	`$G_SCENARIO`	Indicates the base version	varchar(50)	"	"
Yes	`$G_TIME_PROFILE`	The time profile ID is defined in SAP Integrated Business Planning for Supply Chain, in the *Configuration* app.	int	-1	-1

Required by SAP Integrated Business Planning for Supply Chain	Global Variable	Description	Data Type	Default Value for Master Data Task	Default Value for Key Figure Data Task
Optional	$G_TIME_PROFILE_LEVEL	IBP disaggregation level of time profile

This variable is only needed for uploading key figures to IBP from an aggregated time profile level.

If you want to load key figure data from a more aggregated time profile level than the storage time profile level of the key figure, the IBP system can disaggregate the values from the level you specified in this global parameter to the storage level.

Make sure that the value that you set match the required time profile level ID in the time profile that you have defined in the planning area. For example, if you want to load key figures on a quarterly level, set this value to the same as the ID of the quarterly time profile level of the time profile. | int | | 3 |

Required by SAP Integrated Business Planning for Supply Chain	Global Variable	Description	Data Type	Default Value for Master Data Task	Default Value for Key Figure Data Task
Yes	$G_BATCH_COMMAND	Defines how SAP Integrated Business Planning for Supply Chain processes the data loaded in the batch execution. Recognized values are: INSERT_UPDATE updates or adds new data to SAP Integrated Business Planning for Supply Chain. DELETE removes the specified records from SAP Integrated Business Planning for Supply Chain. REPLACE updates key figures that overlap with existing data in the system. The non-overlapping data is not changed.	varchar(50)	'INSERT_UPDATE'	'INSERT_UPDATE'
Yes	$G_LOAD_DATE	The load date of the task	datetime	SYSUTCDATE()	SYSUTCDATE()
No	$G_DEFAULT_TEXT	Default text for the dimension default rows, foreign keys, and primary keys. Also used as a default for missing descriptions.	varchar(1)	'?'	'?'
No	$G_DEFAULT_DATE	Default date for the dimension default rows, foreign keys, and primary keys. Also used as a default for missing dates.	datetime	'1900.01.01'	'1900.01.01'

Required by SAP Integrated Business Planning for Supply Chain	Global Variable	Description	Data Type	Default Value for Master Data Task	Default Value for Key Figure Data Task
Optional	`$G_IBP_FILTER_ID`	Planning filter ID. To use planning filters defined in the *Planning Filters* SAP IBP app, add this global variable to your task and leave the *Value* field empty.	varchar(100)	N/A	
Optional	`$G_IBP_USER_ID`	Business user ID. Authorizations assigned to the given user ID are applied to the selection in SAP IBP.	varchar(50)	N/A	

8.3 Shared Global Variables for Unified Planning Area

All templates for writing key figures to the unified planning area `SAPIBP1` contain the shared global variables described below.

Some of the global variables are also described under Shared Global Variables [page 182], which is the common set of global variables for all other templates.

Global variables contain default values that are used if the value has not been changed. You can change the value of each global variable per your requirements and environment. Certain global variables are used by the SAP Integrated Business Planning for Supply Chain (SAP IBP) applications to process the data after it is loaded. Their descriptions have the prefix 'SAP IBP'. Some global variables do not take inputs, because they are filled with calculated values. Their descriptions contain the suffix '(calculated)'. The following table lists the default values and additional information about each of the shared global variables:

Shared Global Variables for Unified Planning Area

Required by SAP Integrated Business Planning	Global Variable	Description	Data Type	Default Value
Yes	$G_PLAN_AREA	SAP IBP planning area	VARCHAR(50)	'SAPIBP1'

Required by SAP Integrated Business Planning	Global Variable	Description	Data Type	Default Value
Yes	$G_SCENARIO	SAP IBP Version i Note If you want to integrate data into the base version, leave this variable blank (default value).	VARCHAR(50)	''
Yes	$G_BATCH_COMMAND	SAP IBP post-processing batch command (INSERT_UPDATE or DELETE)	VARCHAR(50)	'INSERT_UPDATE'
Yes	$G_TIME_PROFILE	SAP IBP time profile ID (by default, the time profile assigned to the planning area is used)	INTEGER	-1
No	$G_MATE-RIAL_TYPE_1	Material Type 1	VARCHAR(4)	'ROH'
No	$G_MATE-RIAL_TYPE_2	Material Type 2	VARCHAR(4)	'HALB'
No	$G_MATE-RIAL_TYPE_3	Material Type 3	VARCHAR(4)	'FERT'
No	$G_MATE-RIAL_TYPE_4	Material Type 4	VARCHAR(4)	'HIBE'
No	$G_MATE-RIAL_TYPE_5	Material Type 5	VARCHAR(4)	'FHMI'
No	$G_STANDARD_LOAD	Data load with standard settings (calculated)	VARCHAR(3)	
Yes	$G_LOAD_DATE	Load date and time (calculated)	DATETIME	
No	$G_LAST_LOAD_DATE	Last successful load date and time (calculated)	DATETIME	

Required by SAP Integrated Business Planning	Global Variable	Description	Data Type	Default Value
No	$G_LAST_STAND- ARD_LOAD_DATE	Last successful load date and time with standard settings (calculated)	DATETIME	

8.4 Unique Global Variables for Each Template

All SAP Integrated Business Planning for Supply Chain (SAP IBP) data integration tasks contain shared global variables. Additionally, most tasks contain global variables that are unique to the task.

Global variables contain default values which are used if the value has not been changed. As needed, change the value of each global variable per your requirements and environment. The default value and information about each of the unique global variables is shown in the following table.

Task Name	Global Variable	Description	Data Type	Default Value
SOP_KF_Actuals	$G_INITIAL_OR_DELTA	Identifies whether a task is a first (initial) load or a delta load. Recognized values are 'INITIAL' and 'DELTA'.	VARCHAR(7)	'INITIAL'
	$G_DELTA_DAYS	Defines the number of days of data to extract. Default for initial load is 730 days. Default for delta load is 30 days. To change this value, edit the preload script.	INTEGER	730 or 30

Task Name	Global Variable	Description	Data Type	Default Value
	$G_SDATE	Starting date for actuals extraction. It is automatically calculated in the preload script as Starting date for actuals extraction. It is automatically calculated in the preload script as Starting date for actuals extraction. It is automatically calculated in the preload script as `<Today's date> - $G_DELTA_DAYS`.	DATETIME	Calculated in the preload script
SOP_KF_Consumption	$G_LOGICAL_SYSTEM	Logical system value required to call BAPI functions (`BAPI_LOCSRVAPS_GETLIST2` and `BAPI_PDSSRVAPS_GETLIST`). Task returns no data if this value is not correct. Enter the right logical system name for your SAP applications system, following the pattern `'SYSCLNT001'`.	VARCHAR(255)	'APOCLNT800'
SOP_MD_CustomerMaster	$G_PARVW	Use the value defined by your business partner. Allows you to extract customer data for a single value of `KNVP.PARVW`. The default value in the template is 'AG' for sold-to party.	VARCHAR(255)	'AG'

Task Name	Global Variable	Description	Data Type	Default Value
SOP_File_Task	$G_FILE_NAME	Can be used to specify the name of the file that is loaded into SAP Integrated Business Planning. Set the value in the preload script.	VARCHAR(100)	'FileName.csv'
	$G_FILE_DIRECTORY	Can be used to specify the directory of the file that is loaded into SAP Integrated Business Planning. Set the value in the preload script.	VARCHAR(100)	'C:\Directory'
SOP_KF_Inventory	$G_MONTHS_FORECAST	Number of months in the current inventory forecast. 0 is the current month. For example, 18 is 18 months in the future.	INTEGER	18
	$G_MATERIAL_TYPE_1	Extracts inventory data from MARA.MTART for the specified material type	VARCHAR(4)	'ROH'
	$G_MATERIAL_TYPE_2		VARCHAR(4)	'HALB'
	$G_MATERIAL_TYPE_3		VARCHAR(4)	'FERT'
	$G_MATERIAL_TYPE_4		VARCHAR(4)	'HIBE'
	$G_MATERIAL_TYPE_5		VARCHAR(4)	'FHMI'
SOP_MD_LocationMaster	$G_NODETYPE1	Extracts location data for the supply chain network from the specified nodes	VARCHAR(3)	'DC'
	$G_NODETYPE2		VARCHAR(3)	'B'
	$G_NODETYPE3		VARCHAR(3)	No default value
	$G_NODETYPE4		VARCHAR(3)	
	$G_NODETYPE5		VARCHAR(3)	

Task Name	Global Variable	Description	Data Type	Default Value
SOP_MD_LocationProd	$G_MATERIAL_TYPE_1	Extracts inventory data from `MARA.MTART` for the specified material type	VARCHAR(4)	'ROH'
	$G_MATERIAL_TYPE_2		VARCHAR(4)	'HALB'
	$G_MATERIAL_TYPE_3		VARCHAR(4)	'FERT'
	$G_MATERIAL_TYPE_4		VARCHAR(4)	'HIBE'
	$G_MATERIAL_TYPE_5		VARCHAR(4)	'FHMI'
SOP_MD_ProductMaster	$G_MATERIAL_TYPE_1	Extracts inventory data from `MARA.MTART` for the specified material type	VARCHAR(4)	'ROH'
	$G_MATERIAL_TYPE_2		VARCHAR(4)	'HALB'
	$G_MATERIAL_TYPE_3		VARCHAR(4)	'FERT'
	$G_MATERIAL_TYPE_4		VARCHAR(4)	'HIBE'
	$G_MATERIAL_TYPE_5		VARCHAR(4)	'FHMI'
SOP_KF_SalesForecastPrice	$G_DAYS_FORECAST	Number of days that the initial forecast values are based on	INTEGER	183
	$G_MONTHS_FORECAST	Number of future months for which the current sales price will be forecast	INTEGER	18
IBP_MD_PlanningArea	$G_PLANNING_VERSION	Planning version in SAP APO	VARCHAR(3)	'000'
	$G_LANGUAGE	Language	VARCHAR(1)	'E'
IBP_KF_DemandPlanning	$G_PLANNING_UOM	Planning unit of measure	VARCHAR(3)	'ST'

Task Name	Global Variable	Description	Data Type	Default Value
	$G_PERIOD_TYPE	Period type for time selection. Valid values are 'W' for weeks, and 'M' for months.	VARCHAR(1)	'W'
	$G_PLANNING_VERSION	Planning version in SAP APO	VARCHAR(3)	'000'
	$G_PERIODS_FUTURE	Number of periods to be transferred, including the current period	INTEGER	12
	$G_PERIODS_PAST	Number of past periods to be transferred (such as -3)	INTEGER	0
	$G_TIME_PROFILE_LEVEL	Time profile level at which key figure values are loaded. You can get the time profile level from the *Time Profiles* screen in the *Configuration* app.	INTEGER	3 3 corresponds to "week" in time profile 66
IBP_KF_PlannedIndependentRequirements	$G_PLANNING_UOM	Unit of measure of the planning area	VARCHAR(3)	'ST'
	$TMP_TO_PERIOD	Filled in the preload script. Any value you specify for this variable will be overwritten by the preload script.		
	$G_LINEITEMS_PER_PACKAGE	Number of planned independent requirements to be transferred per Web service call. Use this variable to optimize the performance.	INTEGER	1000
	$G_PLANNING_VERSION	Planning version in SAP APO	VARCHAR(3)	'000'

Task Name	Global Variable	Description	Data Type	Default Value
	$G_ATP_CATEGORY	Category of stock receipt, requirement, or forecast. For more information, see the documentation of the `BAPI_PIRSRVAPS_SAVEMULTI` function module in SAP APO.	VARCHAR(2)	'FA'
	$G_LOGICAL_SYSTEM	Logical system. For more information, see the documentation of the `BAPI_PIRSRVAPS_SAVEMULTI` function module in SAP APO.	VARCHAR(10)	'XXXCLNTYYY'
	$G_FROM_DAYS_IN_FUTURE	Number of days (starting from today) that shall be skipped when transferring data. 0 means that the first date to be considered is today.	INTEGER	0
	$G_DAYS_TO_TRANSFER	Number of days to be transferred from SAP Integrated Business Planning to SAP APO	INTEGER	42
IBP_KF_S4_ERP_AddOn	$G_TIMEPROFILEID_ERP	SAP IBP time profile ID for aggregation in SAP ERP or SAP S/4HANA. In the preload script, adapt the following row when changing the default value: `if ($G_TIMEPROFILEID_ERP = ...)`	INTEGER	1

Task Name	Global Variable	Description	Data Type	Default Value
	`$G_TIMEPROFILELEVEL_ERP`	SAP IBP time profile level for aggregation in SAP ERP or SAP S/4HANA. In the preload script, adapt the following row when changing the default value: `if ($G_TIMEPROFILELEVEL_ERP = ...)`	INTEGER	2
	`$G_TIMEPROFILELEVEL_ERP_EXCH_RATE`	SAP IBP time profile level for exchange rate data extraction	INTEGER	4
	`$G_PLANNING_HORIZON`	Planning horizon in the future for open orders revenue The planning horizon can be defined on the time profile level that is used for aggregation, which can be set using the `$G_TIMEPROFILELEVEL_ERP` global variable.	INTEGER	130
	`$G_DELIVERY_STATUS`	Delivery status for open orders revenue	VARCHAR(1)	'A'
IBP_MD_S4_ERP_AddOn	`$G_UNIFIED_BASE_UOM`	SAP IBP unified base unit of measure. It is not generated in case the value is ''.	VARCHAR(3)	'1BU'
	`$G_LANGUAGE`	Language for descriptions	VARCHAR(1)	'E'
IBP_KF_Actuals	`$G_HISTORY_IN_TECHNICAL_WEEKS`	History in number of technical weeks	INTEGER	130
	`$G_HISTORY_START_DATE`	Calculated: History Start Date	DATE	

Task Name	Global Variable	Description	Data Type	Default Value
	$G_COUNTER	Calculated: Internal counter	INTEGER	
IBP_KF_OpenOrders_ERP IBP_KF_OpenOrders_S4	$G_HORIZON_IN_TECHNICAL_WEEKS	Calculated: Planning Horizon in Number of Technical Weeks	INTEGER	130
	$G_PLANNING_HORIZON	Calculated: Last date of Planning horizon	DATE	
	$G_COUNTER	Calculated: Internal counter	INTEGER	
IBP_KF_PlannedPrice	$G_HORIZON_IN_TECHNICAL_WEEKS	Planning horizon in number of technical weeks	INTEGER	130
	$G_HISTORY_IN_TECHNICAL_WEEKS	History in number of technical weeks	INTEGER	130
	$G_TARGET_CURRENCY	Target currency of exchange rates - all planned prices are converted to this currency	VARCHAR(5)	'USD'
	$G_EXCHANGE_RATE_TYPE	Exchange rate type for conversion to planning currency	VARCHAR(4)	'M '
	$G_EXCHANGE_RATE_BEGIN	Exchange rate beginning of search interval	DATE	to_date('00010101','yyyymmdd')
	$G_EXCHANGE_RATE_END	Exchange rate end of search interval	DATE	cast(sysutcdate(),'date')
	$G_EXCHANGE_RATE_BEGIN_INVERSE	Calculated: Exchange rate begin of search interval in inverse format (99999999 - YYYYMMDD)	VARCHAR(8)	
	$G_EXCHANGE_RATE_END_INVERSE	Calculated: Exchange rate end of search interval in inverse format (99999999 - YYYYMMDD)	VARCHAR(8)	
	$G_PLANNING_HORIZON	Calculated: Last date of Planning horizon	DATE	

Task Name	Global Variable	Description	Data Type	Default Value
	$G_HISTORY_START_DATE	Calculated: History Start Date	DATE	
	$G_COUNTER	Calculated: Internal counter	INTEGER	
IBP_KF_SafetyStock	$G_HORIZON_IN_TECHNICAL_WEEKS	Planning Horizon in Number of Technical Weeks	INTEGER	130
	$G_PLANNING_HORIZON	Calculated: Last date of planning horizon	DATE	
	$G_COUNTER	Calculated: Internal counter	INTEGER	
IBP_DDR_to_ERP_AddOn_via_RFC	$G_CURRENT_PERIOD_DATE	Current period date	DATE	
	$G_LINEITEMS_PER_PACKAGE	Approximate line items per package	INTEGER	100000
	$G_PLANNING_HORIZON_IN_DAYS	Future planning horizon in days	INTEGER	365
IBP_to_ERP_PIR_via_RFC	$G_LINEITEMS_PER_PACKAGE	Approximate line items per package	INTEGER	10000
	$G_FIRST_SELECTION_WEEK	First selection week (before or after the current week)	INTEGER	0
	$G_LAST_SELECTION_WEEK	Last selection week (before or after the current week)	INTEGER	105
	$G_VERSION_ACTIVE	Indicator: Version active	VARCHAR(1)	'X'
	$G_VERSION_NUMBER	Version number	VARCHAR(2)	'00'

Task Name	Global Variable	Description	Data Type	Default Value
	$G_REQ_TYPE	Requirements type	VARCHAR(4)	'VSF'
				If left empty, requirements type is defined based on the main planning strategy as of SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning version 1.0 SP18 and SAP ERP, supply chain integration add-on for SAP Integrated Business Planning version 1.0 SP21.
	$G_REQ_PLAN_NUMBER	Requirements plan number	VARCHAR(10)	''
	$G_AGGREGATION_UOM	Unit of measure for aggregation of the consensus demand quantity key figure in SAP IBP	VARCHAR(3)	'1BU'
	$G_PROCESS_PIR	Processing planned independent requirements automatically in the add-on	VARCHAR(1)	'X'
	$G_DATE_TYPE	Date type (day, week, month, interval)	VARCHAR(1)	'2'
	$G_JOB_GUID	GUID of task or process	VARCHAR(32)	gen_uuid()
	$G_LOG_HELPER	Helper field for writing log entries in preload script	VARCHAR(50)	''
	$G_DELETE_OLD	Flag to delete planned independent requirements for the location material from SAP S/4HANA or SAP ERP before integrating new records	VARCHAR(1)	'X'

Task Name	Global Variable	Description	Data Type	Default Value
IBP_DDR_TO_S4_BUF FER_INTEGRATION	$G_BUFFER_INTEGRA TION_IN_DAYS	Number of days to transfer demand-driven replenishment buffers to SAP S/4HANA	INTEGER	30
IBP_DDR_to_S4_Pos ition_Information	$G_HORIZ_DAYS_PLU S	Period for average daily usage calculation in future days	INTEGER	30
	$G_HORIZ_DAYS_MIN US	Period for average daily usage and lead time calculation in past days	INTEGER	0
	$G_PQR_VALUE	Value from PQR analysis	VARCHAR(4)	''
	$G_HORIZON	Number of days to be considered for transfer of data	INTEGER	30
IBP_DDR_to_S4_Siz ing_Information	$G_HORIZ_DAYS_PLU S	Period for average daily usage calculation in future days	INTEGER	30
	$G_HORIZ_DAYS_MIN US	Period for average daily usage and lead time calculation in past days	INTEGER	0
	$G_PQR_VALUE	Value from PQR analysis	VARCHAR(4)	''
	$G_HORIZON	Number of days to be considered for transfer of data	INTEGER	30
IBP_PPDS_to_file_ IBP_TimeProfile	$G_FILE_NAME	Target file for time profile extraction from SAP IBP	VARCHAR(100)	'IBPPPDS_TIME_PRO-FILE.csv'
IBP_PPDS_from_S4_ Res_CapInfo	$G_PROPAGATION_RA NGE	Propagation range for SAP S/4HANA Advanced Planning	VARCHAR(20)	'<Name of your propagation range set in S/4HANA>'

Task Name	Global Variable	Description	Data Type	Default Value
	$G_FILE_NAME	File where the SAP IBP time profile selection is stored	VARCHAR(100)	'IBPPPDS_TIME_PRO-FILE.csv'
IBP_PPDS_from_S4_Orders	$G_PROPAGATION_RANGE	Propagation range for SAP S/4HANA Advanced Planning	VARCHAR(20)	'<Name of your propagation range set in S/4HANA>'
	$G_ATP_CATEGORIES_STR	ATP categories for STRs maintained as a list	VARCHAR(200)	'AG, AH, BF'
	$G_ATP_CATEGORIES_PROD	ATP categories for production orders maintained as a list	VARCHAR(200)	'AA, AB, AC, AD, AI, AJ, AL, NC'
	$G_ATP_CATEGORIES_PROC	ATP categories for procurement receipts maintained as a list	VARCHAR(200)	'AG, AH, BF'
	$G_FILE_NAME	File where the SAP IBP time profile selection is stored	VARCHAR(100)	'IBPPPDS_TIME_PRO-FILE.csv'

9 Extensibility Information

This chapter gives an overview of the development and extension options you have for the integration of SAP Integrated Business Planning with SAP Business Suite systems using SAP Cloud Integration for data services.

This documentation does not replace the documentation in the Help Center for SAP Cloud Integration for data services at https://help.sap.com/viewer/product/ SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES (especially the chapters *Data Flows*, *Transform Operations*, *Scripts and Functions*). Some of the topics are covered in the SAP Data Services Agent Guide at https://help.sap.com/viewer/product/ SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES, for example in the chapter *Considerations for running ABAP programs*.

SAP Integrated Business Planning is a planning tool with a built-in extension concept based on a flexible data model. You can add new objects, attributes, and key figures to the planning area. To extend the planning model in SAP IBP or to fill existing attributes and/or key figures with content from a source system, you can extend the mapping content in SAP Cloud Integration for data services. Depending on your use case, there are different options.

Field extensions are quite easy to handle if they belong to a table or to an extractor that is already used as a data source. You can also add fields from other tables or define additional selection conditions. Most calculations on field level are easy, but there are also more complicated use cases, for example currency conversions and aggregations. In many cases this type of extension is difficult to solve using existing tables and extractors as the only data source. It is often better to create customer-specific staging tables containing the data to be replicated to SAP IBP in a fitting format and to fill those tables using customer-specific reports. As an alternative, you can define your own extractors or use the custom ABAP transform to read data source code based without the need of persisting the data in staging tables.

If you use the SAP S/4 HANA, supply chain integration add-on for SAP IBP or the SAP ERP, supply chain integration add-on for SAP IBP to read data from an S/4HANA or ERP system and transfer it to SAP IBP, you have the option to extend the used extractors with customer-specific fields. You can also modify selection criteria and output data using the implementation of a Business Add-In (BAdI). For more information, see the administrator's guides of the add-ons at https://help.sap.com/ibp_s4hana_addon or https://help.sap.com/ ibp_erp_addon.

If you need data from different systems, the best approach is usually to consolidate this data

If you need data from different systems, the best approach is usually to consolidate this data in a single system, like SAP BW or SAP APO before sending it to the SAP IBP system. This is even more important if the data needs to be consolidated, including ID mappings, for example.

9.1 Field Extensions

In SAP Integrated Business Planning additional attributes can be added to the model of the planning area.

In SAP Business Suite source systems fields can be added to tables and structures using append structures. There are several options for filling these additional fields with data. For example, in SAP S/4HANA systems,

you can create screen extensions to master data maintenance screens. You can also use BAdIs or other extension concepts to fill additional data. An existing or newly created field in the SAP Business Suite source system is mapped to an existing or newly created attribute or key figure in the SAP Integrated Business Planning system.

Fields from Tables or Extractors Already in Use

1. If the field is newly created in the source system or in SAP Integrated Business Planning you might need to reimport the corresponding object to the datastore to add the newly created field.
2. Open the task you want to extend in edit mode.
3. Edit the dataflow you want to extend.
4. Open the first transform assigned to the source table.
5. Drag and drop the field you want to add to the dataflow from the input fields list to the output structure.
6. You can optionally edit the newly created field by clicking the *Edit* button of the newly created output field entry. You can change the name, edit the data type, and assign a different description.
7. If you need a more complex mapping, you can set the cursor on the output field, click *Mapping* and edit the content of the field mapping. For more details, see Calculations in Data Flow Transforms [page 206].
8. Repeat steps 4 to 7 for all transforms on the way to the target table.
9. If the source field and the target field do not have the same name, data type, or length you need to switch from one format to the other. It is best not to perform this mapping in the first transform after the source table nor in the last transform before the target table. This is because you might need to replace the source or target table by another one and then it is easier to reassign the fields if they have the same names, format, and a 1:1 mapping. For example, in the templates for reading key figures for the unified planning area, the mapping from the source name and format to the target name and format is done between the last ABAP query before the ABAP endpoint and the follow-on non-ABAP query, so between the mapping content executed in the source system and the one executed on the agent.

Fields from Additional Tables

To add fields from a table that is not yet a source table of a dataflow, you have the following options:

- Add the corresponding table as a source table and join it with the existing tables or extractors. How the source tables and/or extractors are selected depends on the definition of the dataflow. To push a join of several tables down to one selection statement in the source system, some prerequisites must be fulfilled:
  - Source tables are all of the type *Table* (tables and/or database views) and no extractors are involved.
  - The join is defined in an ABAP query or an ABAP aggregation, so that the selection is executed in the generated ABAP report.
  - The join condition only contains inner joins (left outer joins lead to select single statements for the left outer joined table).
  - The field mapping of the transform with the join condition and all its sources does not contain any functions that cannot be pushed down to the database, like `ifthenelse( )` or `decode( )`.

  On the other hand there are situations not avoiding the pushdown to one select statement:
  - Define joins in several directly linked ABAP queries or ABAP aggregations

- Define ABAP query transforms in-between the source table(s) and the join transform(s)
- Create a database view in the source system, which reads data from one or several existing tables and the new one and replace the existing source table(s) by the database view. For more information, see Database Views [page 208].
- Use the lookup() function to read the table field.
  There are three different values for the cache specification, which lead to a different behavior of the cache:
  - NO_CACHE this leads to a select single statement on the lookup table per entry of the dataflow table. This is the best choice if the same lookup value in most cases is only read once or twice and only a minor part of the lookup table entries needs to be read.
  - PRE_LOAD_CACHE reads the whole lookup table into a cache table and reads from there. This is the best choice if the majority of the lookup table entries is needed. If some of the selection conditions are defined using fixed values, these selections are pushed down to the database. This is not the case for global variables.
  - DEMAND_LOAD_CACHE leads to a cache table that is filled by several select single statement. Every new combination of selection fields leads to a select single, but then the selection fields and the lookup field value are stored in a cache table and reused for the same selection field combination. This is the best choice if the number of selection field combinations is low and only a small part of all entries of the lookup table is read.

9.2 Select Conditions

Hard-Coded Conditions

Transforms of the type *Query* or *ABAP Query* have the tab *Filter* where you can define filter conditions using constant values, for example numbers or strings. An example could be:

MARA.NTGEW > 0 and

MARA.MTART in ('ROH', HALB', 'FERT', 'HIBE', 'FHMI')

On the *Join* tab of queries, aggregations, ABAP queries, and ABAP aggregations you can define join conditions using fixed values, for example:

T006A.MSEHI = MARA.MEINS and

T006A.SPRAS = 'E'

A third place to define selection conditions using fixed values is the lookup() function, for example:

lookup(ERP..T006A,MSEHL,'','PRE_LOAD_CACHE',MSEHI,MARA.MEINS,SPRAS,'E')

Using fixed values is the only way of reducing the number of entries read with the lookup() function in PRE_LOAD_CACHE mode. The above statement stores all English descriptions of units of measurement in the cache table. For more details, see the subchapter *Lookup Tables*.

Pros: Easy to implement, high probability to reduce the amount of selected data

Cons: Intransparent, the conditions are hidden in field mappings or join conditions of transforms within a dataflow

Filter Using Global Variables

Instead of using hard-coded selection conditions as described above you also can use global variables. The examples from above could also look like this:

`MARA.NTGEW > 0` and

`MARA.MTART` in (`$G_MATERIAL_TYPE_1`, `$G_MATERIAL_TYPE_2`, `$G_MATERIAL_TYPE_3`, `$G_MATERIAL_TYPE_4`, `$G_MATERIAL_TYPE_5`)

`T006A.MSEHI = MARA.MEINS` and

`T006A.SPRAS = $G_LANGUAGE`

`lookup(ERP..T006A,MSEHL,'','PRE_LOAD_CACHE',MSEHI,MARA.MEINS,SPRAS,$G_LANGUAGE)`

If the values of the global variables correspond to the fixed values from above, the results will be the same. Only the cache table of the lookup function will contain more data. For more information, see the subchapter *Lookup Tables*. The main differences between using global variables instead of fixed values are the following:

- Global variables can be overwritten when running or scheduling a task. For example, if you change the values of the global variables `$G_MATERIAL_TYPE_2` to `$G_MATERIAL_TYPE_5` to ' ', only materials with material type 'ROH' are selected.
- Global variables can be set in the preload script of the task. You even can do calculations using while loops there. If a global variable is set in the preload script, this overwrites the value set by the input parameters.

Pros:

- The selection is more flexible because it can be influenced by input parameters or calculated values.
- More transparency as the global variables are visible on task level

Cons:

- Adding a new value to a list of possible values is more complicated, because you have to define a new global variable and change the corresponding filter conditions to be able to use the new variable.
- The higher flexibility of the selection also means a higher risk of unwanted behavior or even data loss. For more information, see Batch Command 'REPLACE' [page 228].
- You still need to adapt all filter conditions in different places within transforms of task dataflows.

Planning Filters Coming from SAP IBP

Filter conditions set in the *Planning Filters* app are considered when transferring key figure data from SAP IBP via SAP Cloud Integration for data services. You can use planning filters to restrict data integration.

> **i Note**
>
> Planning filters cannot be applied for master data integration.

When using planning filters in data integration, SAP IBP sends the filter ID and the permission filter information to SAP Cloud Integration for data services, where it is applied for tasks that include the necessary global variables.

To apply planning filters, do the following:

1. In SAP IBP, specify filter criteria and corresponding user groups for your planning areas using the *Planning Filters* app.
2. In SAP Cloud Integration for data services, add the following global variables to the tasks in which you want to apply filters, and leave their *Value* fields empty:
   - $G_IBP_FILTER_ID
   - $G_IBP_USER_ID

   If needed, define additional filter criteria in the data flow. However, it is recommended not to set the same conditions in SAP Cloud Integration for data services that you've already specified in SAP IBP. If you use templates, it is recommended to remove duplicated filters from the QueryAndFilter step.

> **i Note**
>
> Data integration using planning filters requires SAP Cloud Integration for data services version 1.0.11 patch 40 or later, and Data Services Agent version 1.0.11 patch 40 or later.

For more information about the *Planning Filters* app, search for *Planning Filters (New)*, and for user permissions, search for *User Authorizations and Permissions* under SAP Integrated Business Planning for Supply Chain on the Help Portal at http://help.sap.com/ibp.

Replace Tables by Database Views

Instead of defining selection conditions within dataflows, you also can define database views with selection conditions and use them as source tables to replace the underlying tables.

Using transaction SE11 in your SAP S/4HANA or SAP ERP system you could create a database view ZV_MARA based on table MARA with the following selection conditions:

Table	Field Name	Operator	Comparison Value	AND/OR
MARA	MTART	EQ	'ROH'	OR
MARA	MTART	EQ	'HALB'	OR
MARA	MTART	EQ	'FERT'	OR
MARA	MTART	EQ	'HIBE'	OR
MARA	MTART	EQ	'FHMI'	

Then you could import the database view as a table to the datastore and replace the source table MARA in all relevant dataflows by source table ZV_MARA.

Pros:

- Selection conditions are defined at one central place and can easily be extended, if needed.
- Avoids inconsistencies between filter conditions in different dataflows
- Potentially smaller cache table of lookup() function in PRE_LOAD_CACHE mode

Cons:

- An implementation in the source system is needed.
- Initial implementation effort might be a little higher.
- You cannot use global variables to dynamically influence the selection. Global variables can only be used to filter the entries afterwards.

Lookup Tables

The lookup() function selects one field from one source table. Every lookup table has several selection conditions. Every condition consists of a comparison of a field with some other value. The comparison operator is always *Equal*. The compared value can be an input field of the current transform, a constant value, a global variable, a function, or nested functions.

If the cache mode is NO_CACHE or DEMAND_LOAD_CACHE, table entries are selected using a select single statement. In case of DEMAND_LOAD_CACHE the entries already selected are cached.

If the cache mode is PRE_LOAD_CACHE, the whole lookup table is cached. The only exception are selection conditions with hard-coded values. For example, with the following lookup function all English material descriptions are selected and stored in the cache table:

Lookup(MAKT.MAKTX, MAKT.MATNR, MARA.MATNR, MAKT.SPRAS, 'E')

If you use a global variable $G_LANGUAGE with value 'E' instead, all material descriptions in all languages are read and stored in the cache table:

Lookup(MAKT.MAKTX, MAKT.MATNR, MARA.MATNR, MAKT.SPRAS, $G_LANGUAGE)

If you want to use the cache mode PRE_LOAD_CACHE, but you do not want to select the whole table it might be useful to create a database view instead. For example, if you only want to select English and German descriptions, you could create a database view ZV_MAKT in the source system that is based on table MAKT and has the following selection conditions:

Table	Field Name	Operator	Comparison Value	AND/OR
MAKT	SPRAS	EQ	'E'	OR
MAKT	SPRAS	EQ	'D'	

If you use this database view instead of table MAKT in the lookup() function using cache mode PRE_LOAD_CACHE the cached table will only contain English and German descriptions.

You could even go one step further and add the table MARA to the database view by the foreign key relationship of table MAKT. Then the selection conditions could look like this:

Table	Field Name	Operator	Comparison Value	AND/OR
MAKT	SPRAS	EQ	'E'	OR
MAKT	SPRAS	EQ	'D'	AND
MARA	MTART	EQ	'ROH'	OR
MARA	MTART	EQ	'HALB'	OR
MARA	MTART	EQ	'FERT'	

As a result, the database view and thus the cache table would only contain English and German descriptions of the materials with the material group 'ROH', 'HALB', or 'FERT'.

You are also more flexible defining selection conditions because you can choose the operator and the AND/OR condition.

9.3 Calculations in Data Flow Transforms

When assigning a value to a target field in a transform, you can not only assign a source field, a fixed value, or a global variable to it, but you can also use comparison operators, like =, !=, <= and combine them using operators like AND, OR, and NOT.

You can also do calculations using operators like +, -, *, /, % and concatenate values by using ||, and so on. All these operators and calculations can also be used in preload and postload scripts. For a complete list of supported operators, look for *Operators* in the Help Center for SAP Cloud Integration for data services at https://help.sap.com/viewer/product/SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES.

You can also use functions for mapping a value to a target field and in scripts. Functions can have zero, one, or more parameters and a returning parameter providing a value. For a complete list of available functions, look for *Descriptions of Functions* in the Help Center for SAP Cloud Integration for data services at https://help.sap.com/viewer/product/SAP_CLOUD_PLATFORM_INTEGRATION_FOR_DATA_SERVICES.

This documentation does not describe in detail which functions are available in the various types of transforms. You can see which ones are available when you open the *Mapping* tab of a data flow transform. To do so, mark an output field of a dataflow transform and click on the *Mapping* tab of the *Transform Details*. You find one or more dropdown boxes for different categories of transforms that can be used in the current transform type. If you choose one of the functions in edit mode, you can expand the function help to get a short description of the function and an overview of the available parameters. Please note that the description of the functions in the Help Portal contains more detailed information. You can specify the content of the parameters. If you click *OK*, the function is added to the mapping content of the output field at the cursor position.

Functions in non-ABAP Queries

This transform type has the most categories of functions available: *Conversion*, *Cryptographic*, *Date*, *Lookup*, *Math*, *Miscellaneous*, *String and Validation*. The only category that is not available here is *Aggregation*. Besides that, only the function *sy()* from ABAP queries is missing.

Functions in ABAP Queries

The available categories for functions are: *Date*, *Lookup*, *Math*, *Miscellaneous*, and *String* and the number of functions available per category is significantly smaller. The only function that is available here and not in query transforms, is the function *sy()*. It can be used to read system variables from the source system.

Functions in Aggregations and ABAP Aggregations

Aggregations and ABAP aggregations only have the function category *Aggregations.* As a consequence, only those aggregation functions are available that are not available in the query transform types: `AVG`, `COUNT`, `COUNT_DISTINCT`, `MAX`, `MIN`, `SUM`.

The mapping logic on aggregations is quite simple. Every field in the target structure either must be assigned to a source field that is a grouping condition or to an aggregated value of a source field that is not a grouping condition.

9.4 Calculate Global Variables in Preload Scripts

All the functions available for non-ABAP queries can also be used in preload and postload scripts.

Additionally, the following keywords are supported:

- `BEGIN`
- `CATCH`
- `ELSE`
- `END`
- `IF`
- `RETURN`
- `TRY`
- `WHILE`

These enable you to do more complicated calculations within scripts including loops.

Variables you want to read or write in scripts must be defined as global variables. If a global variable is set to a specific value in the preload script, this will overwrite the default value and a value that was set by overwriting the default value in the input when starting a task. In the templates used for writing key figures to the unified planning area this is handled the as follows:

The global variables set to a calculated value in the preprocessing script contain a suffix 'Calculated:' in the description. Like this the end user starting a job is aware that it does not change the behavior of the task to set a different value for those variables. Additionally, the preload script places a warning if a calculated value has been set by the input.

For non-calculated global variables the preload script compares the input value with the default value and writes to the trace log if the value is the default value or a changed one. If you change the default value of a global variable, you should adapt the default value in the preload script accordingly.

9.5 Database Views

In datastores of the type *Business Suite System* and *BW Source* you can use the functions *Import Objects* and *Import Object by Name* to import metadata for extractors and tables from the source system.

You can also import database views and projection views using the function *Import Object by Name* and choosing the type *Table*. Database views and projection views can be displayed and maintained in transaction SE11 in the source system. There are four types of views available in Business Suite systems: database views, projection views, help views, and maintenance views. The first two can be used as a source in a datastore, the other two cannot.

To search for database or projection views in the business suite system, proceed as follows:

1. Start transaction SE11 and open the value help for the field *View Name*.
2. Click on *Information System* and then *All Selections*.
3. Select *DB Views* and *Projection Views*.
4. Add additional selection conditions, if needed, and click *Execute*.

Projection views are based on one database table. The purpose of projection views is to reduce the list of fields to those table fields the user of the view needs.

You can achieve the same by defining a database view. It is also based on one basis table. But you can add additional tables by foreign key relations of the base table.

You could, for example, join the table MARC (Plant Data for Material) with table MARA (General Material Data), and T134 (Material Types) as follows:

1. Start transaction SE11 and fill field *View Name* with your view name, for example, *ZV_MARC_ETC*.
2. Select *View* and then *Create*.
3. Choose the view type *Database View* and click *Copy*.
4. Fill in the short description, for example *Plant Data for Materials with Related Data*.
5. Enter **'MARC'** in the first row of the table *Tables*, mark this first row in table and click on button *Relationships*.
    You see a list of available foreign key relations for table MARC, several of them having MARC in the first column and MARA in the second one. You can open the details of every entry by double-clicking it.
6. Find the entry that assigns the fields MARC-MANDT to MARA-MANDT and MARC-MATNR to MARA-MATNR, mark the checkbox for this entry and click *Copy*.
7. Mark the newly created second row **'MARA'** in table *Tables* and click *Relationships*.
8. Mark the entry MARA|T134 and click *Copy*.
9. Click the tab *View Flds*, click on *Table fields* and then choose the table MARC.

10. Mark all fields you want to take over from table `MARC`, for example.`MANDT`, `MATNR`, `WERKS`, `LVORM`, and the click *Copy*.
11. Repeat this procedure for the table *MARA* and the desired fields (for example, `ERSDA` and `ERNAM`) and also for the table `T134` and the desired fields (for example, `MTART` and `KZMPN`).
12. Save and activate your changes.

Now you've created one database view that selects the combination of main table `MARC` with the tables `MARA` and `T134` from foreign key relations.

Please be aware that the three tables are joined with an inner join condition. This means that entries from the main table are not selected if the corresponding foreign key tables do not have a fitting entry. For example, if an entry in table `MARA` has an entry in field `MTART` (material type), which does not exist in table `T134` (or is empty), this entry is not selected from the table `MARA`. The corresponding entries from table `MARC` with the same material number are also excluded from the selection.

After you have activated the database view you can import it to the datastore created for this source system as follows:

1. Log on to the fitting organization of the SAP Cloud Integration for data services server and click on the tab *DATASTORES*.
2. Select the fitting datastore in the list on the left of the screen and click on the tab *Configuration*.
3. If the default configuration is not the one for the creation system of the database view, proceed as follows:
    1. Click on the creation system of the database view and click *Save As Default Configuration*.
    2. Click *Save* to persist the change
    3. Click *Test Connection with Default Settings* to check if the technical connection is working correctly.
4. Click *Tables* and then *Import Object by Name*. Choose the type *Table* if you haven't done this yet.
5. Enter the name of the database view in the field *Name*.
6. Check if the database view has been added to the list of tables. If not, there might be several reasons, for example
    - The database view does not exist or is inactive in the datastore default configuration system.
    - The technical connection to the source system is not working
    - The RFC user is missing some authorizations. For more information, see Authorizations for the SAP Business Suite RFC user [page 218] and Troubleshooting [page 230].

If the database view is available in the datastore, you can edit all dataflows using the basis table or the combination of tables with fitting join conditions and selection criteria and replace them by the database view.

9.6   Extractors

There are several ways of enhancing extractors in the source system.

If you use the SAP S/4 HANA, supply chain integration add-on for SAP IBP or the SAP ERP, supply chain integration add-on for SAP IBP to read data from an S/4HANA or ERP system and transfer it to SAP IBP, you have the option to extend the used extractors with Customizing includes. You can also modify selection criteria and output data using the implementation of the Business Add-In (BAdI) `/IBP/ BADI_ES_ETS_SELECTION`. For more information, see the administrator's guides of the add-ons at https:// help.sap.com/ibp_s4hana_addon or https://help.sap.com/ibp_erp_addon. The guides describe the options

you have to fill or modify these fields. For extractors defined in the add-on for SAP IBP, we recommend to use this approach.

If you just want to manipulate existing fields of other extractors, you can implement the BAdI RSU5_SAPI_BADI. For more information about this BAdI, start transaction SE18, enter the BAdI name RSU5_SAPI_BADI, click *Display* and then *Documentation*. To add customer-specific fields, you can also create an append to the selection structure or the underlying table. You can then use the BAdI or any other extension technique to fill these additional fields. It might be necessary to make the additional fields visible using transaction RSA6 afterwards.

Alternatively, you can also create customer-specific extractors from scratch. This could be useful, for example in the following situation: If you want to read data from a source table but you need to manipulate some of the field values in the source system you can create a table-based extractor and then implement BAdI RSU5_SAPI_BADI to perform the required field manipulations.

Creating a function module based extractor is a little more complicated, as you also need to create the underlying function module.

9.7   Staging Tables

Due to the complexity of the dataflows in SAP Cloud Integration for data services, extensive extensions to the selection and mapping of data in the source system can quickly rise to a level that can no longer be maintained without extensions in the source system(s).

This is especially true for complicated calculations, like time-dependent currency conversions or in cases where the easiest way of reading data is not based on table access, but by calling some source code, like function modules or classes. One way of simplifying the mapping logic in SAP Cloud Integration for data services is to use staging tables in the source system that have a structure that is very closely related to the input staging tables of the IBP system. These tables can be filled and updated using planned jobs in the source system by running customer created reports, for example. The staging tables can then easily be mapped to the IBP inbound staging tables.

9.8   RFC-Enabled Function Modules

SAP Cloud Integration for data services together with the latest version of the Data Services Agent supports direct calls of Remote-Function-Call enabled function modules.

In datastores of the type SAP Business Suite Applications, you can load three types of interface definitions, tables, extractors, and functions. Functions can only be loaded by name, and not by the hierarchy search. SAP Cloud Integration for data services only supports RFC-enabled function modules, thus, local ones are not supported. The function modules cannot be added to the ABAP part of the data flow.

RFC Calls Compared with Web Services

However, this approach has some differences compared to web service calls. The RFC call from SAP Cloud Integration for data services is always synchronous, so if the processing time is too long, the call times out. The setting in the datastore for background processing only applies to table and extractor access, not to the functions.

To call an RFC-enabled function module, it used to be necessary to define a web service for the function module and to create a binding for it in the `SOAMANAGER` transaction. Then, you had to create an extra datastore of the type `SOAP Web Service` and load the WSDL file there. This is not required any longer in case the web service is based on an RFC-enabled function module. You can load the function module definition to the datastore that you have created for the integration between SAP S/4HANA or SAP ERP and SAP IBP.

There are also some limitations on the supported function modules. For example, no nested tables are supported. It is not possible to define a table-like parameter with `TABLES TYPE <table type>` either. However, `TABLES` parameters defined with `LIKE <structure name>` are supported and are even faster than parameters of type `IMPORTING`, `CHANGING` or `EXPORTING TYPE <table type>`. Please note that parameters defined with `TABLES` always have a header line with the same name. Therefore, if you want to find out whether a table is initial, use `IF <table name>[] IS INITIAL`.

Another difference is that an RFC-enabled function module cannot be the target of a data flow, but it always has to be an intermediate step. If you would like to rework an existing data flow calling a web service so that it calls the function module directly instead, you can only do so if the web service is an intermediate step, and not the target. Otherwise, you need to define a new data flow.

The handling of the function call is also slightly different from the web service call. If the function module call technically fails, the output structure of the function module in SAP Cloud Integration for data services contains an extra parameter, `AL_RFC_RETCODE`. If its value is `RFC_OK` after the call, everything is fine, but if not, make sure that an exception is thrown to make the overall data flow fail. You also have the option to throw exceptions depending on the output of the function module, and do so if the output contains error statuses or error messages.

Templates Calling RFC-Enabled Function Modules

There are multiple SAP Cloud Integration for data services templates available that call RFC-enabled function modules in SAP ERP, supply chain integration add-on for SAP Integrated Business Planning and SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning.

Template for Integrating Planned Independent Requirements

This template is for the integration of planned independent requirements (PIRs) into SAP S/4HANA or SAP ERP. It contains one data flow that selects a key figure from the SAP IBP system, then divides the data into packages making sure that all data relevant for one location product are sent together. Then, the function module is called for each package. The function module stores the data in a staging table and, optionally, immediately runs the processing of the data.

In case errors occur during data processing, error messages are sent back to SAP Cloud Integration for data services and are displayed in the job log. In case the function call has a technical error or the storing of the data in the staging table fails, the data flow throws an exception, and the data flow gets an error status. You can adapt the behavior of the output error handling to your needs.

The called `/IBP/ETS_PIR_IN_RFC` function module is implemented in a performance-optimized way. The two table-like parameters are defined as `TABLES` parameters:

- `CT_PIR_IN LIKE /IBP/S_ETS_PIR_IN`
- `ET_MESSAGE LIKE BAPIRETURN1`

It is important to define the `TABLES` parameters with `LIKE <structure name>` instead of `TYPE <table type>`, as SAP Cloud Integration for data services is not able to handle the latter version correctly. `TABLES` parameters are always handled as `CHANGING` parameters, so they can be used for importing and exporting data, however, in most cases, they are only needed for one of these use cases. The `CT_PIR_IN` parameter is used to import the PIR information. In order to reduce the data volume returned to SAP Cloud Integration for data services, the `CT_PIR_IN` table is cleared after the data has been stored. From a technical point of view, it is a changing parameter, but it is only used for importing data.

> **i Note**
>
> Although you can also define table-like parameters with `IMPORTING`, `CHANGING`, or `EXPORTING TYPE <table type>`, it leads to bigger data volumes being transferred, and therefore it is not recommended in case of big data volumes.

For more information, see IBP_to_ERP_PIR_via_RFC [page 97].

Template for Integrating DDMRP-Relevant Data to an SAP ERP System

The `IBP_DDR_to_ERP_AddOn_via_RFC` template is relevant for you if you use demand-driven replenishment and integrate data to an SAP ERP system.

It can also be used as a technical template for defining your own tasks with data flows based on your own function modules. The template contains several data flows that read data from key figures and master data and sends them to different function modules. The first data flow, `IBP_DDR_to_ERP_DDPrdLoc`, is a good example of a simple data flow that does not include packaging of the data, contains only one function call, and has simple error handling. At the moment, the underlying function modules are not performance-optimized as described above. If possible, use `TABLES LIKE <structure name>` parameters instead.

To adapt one of the above data flows to call a different function module, do the following:

1. Delete the `CallRFC` transform, and replace it with a new one of type *Web Service or Function Call*.
2. Name the new transform **CallRFC** as well.
3. Recreate the arrows from `PrepareRFC` to **CallRFC** and from **CallRFC** to `GetRFCOutput`.
4. Open the `PrepareRFC` transform.
5. On the output side, load the function module interface.
6. Make sure you map all the input to the function module interface. Note that this includes setting iteration rules for all deep substructures of the function module and also of the root node in case of several function calls.
7. If you have `EXPORTING`, `IMPORTING`, or `CHANGING` tables that will stay empty, delete them from the output structure of the `PrepareRFC` transform.
8. Open the **CallRFC** transform, and load the function module definition again.
9. Map the first-level nodes of the `PrepareRFC` transform to the corresponding nodes of **CallRFC**. Map only parameters and tables, but there is no need to map all fields of the tables.
10. Adapt all other transforms of the data flow to your needs so that the required data is read from SAP IBP with the relevant filters, while the error handling and printing of the output are also set correctly.

For more information, see IBP_DDR_to_ERP_AddOn_via_RFC [page 113].

Templates for Data Relevant for Production Planning Integration

The templates are relevant for you only if you use production planning integration for key figures and integrate data to SAP S/4HANA Advanced Planning (Production Planning and Detailed Scheduling (PP/DS)).

The templates contain the following tasks that either read or write data to SAP IBP:

- A helper task (`IBP_PPDS_to_file`) for reading the SAP IBP time profile to a file, which is then used when extracting data from SAP S/4HANA Advanced Planning.
- The read tasks (`IBP_PPDS_to_S4`) contain data flows that select key figures from the SAP IBP system, then divide the data into packages, making sure that all data relevant for one location product are sent together. Then, the relevant function module is called for each package and starts processing the records in SAP S/4HANA. In case errors occur during data processing, error messages are sent back to SAP Cloud Integration for data services and are displayed in the tasks `Trace log` and `Error log`.
- The write tasks (`IBP_PPDS_from_S4`) are used to call RFC-enabled function modules in SAP S/4HANA Advanced Planning to write key figure data to SAP IBP.

For more information, see Key Figure Templates for Production Planning Integration [page 151].

9.9 Web Services Based on RFC-Enabled Function Modules

In addition to the staging tables described above, there are other ways to achieve a source code based access to data in the source system. One is to generate a web service based on an existing or customer-created RFC-enabled function module, for example a BAPI.

The web service must be created as a separate datastore in SAP Cloud Integration for data services.

As the web service cannot be a source or target datastore of a task, define the task with a file format group as source datastore and SAP IBP as a target datastore. The web service can then be added as an intermediate transform in the dataflow. The drawback of this approach is that the web service is only called once in the dataflow and the expectation is that all required data within the dataflow is given to the Data Services Agent at once. Therefore, processing in batches is not available as it is the case for tables, extractors, and Custom ABAP Transforms.

The web service can also be called when SAP IBP is the source and the file format group is the target. This method can be used to write data back to the business suite system. In this scenario it might be possible to divide the data into batches in SAP Cloud Integration for data services before calling the web service. This would mean that SAP IBP outbound the processing in batches might be possible. For more information, see IBP_KF_PlannedIndependentRequirements [page 141].

A third use case for using a web service based on an RFC-enabled function module is to read data from one system (using tables, extractors, etc.), then call the web service of another system, which has a changing table parameter or a combination of input and output table parameters, in order to manipulate or enhance the data from the first system by data from the second one.

> i Note
>
> SAP Cloud Integration for data services does not support nested tables in the interface structure of the web service.

> **i Note**
>
> If possible, use direct RFC calls instead of web services based on RFC-enabled function modules. For more information, see RFC-Enabled Function Modules [page 210].

9.10 OData Services

Similar to web services OData services cannot be a source or target of a dataflow. They can only can be defined as an intermediate transform.

For SAP S/4HANA there are many OData services available that can be used to read or write data. Additionally, it is also possible to generate OData services for ABAP Core Data Services views (CDS-views).

Please also note that selecting data in the source system and rendering the OData xml message out of the selected data can consume a lot of resources in the source system. The consumption of memory in the Data Services Agent might also greatly increase when receiving an OData message with a big amount of selected data. The same is true for writing data to a business suite system. However, you have some options to ensure that data is written in batches.

If you plan to use an OData service for integration, please check if the performance is sufficient for the expected data volume in the productive environment.

9.11 Custom ABAP Transforms

Custom ABAP transforms consist of several parts:

* The definition of the source code of a form routine stored on the Data Services Agent
* A list of output fields
* A list of report parameters

The list of output fields defines the structure of the output table of the form routine. The list of report parameters defines the import parameters of the generated report.

When a dataflow with a Custom ABAP transform is executed in *Generate and Execute* mode, the system generates a report that contains the report parameters and several form routines. One of these routines has the source code stored on the Data Services Agent and an output table containing the fields from the list of output fields.

The pros and cons of a Custom ABAP Transform versus a web service based on RFC-enabled function module are the following.

Pros:

* Data can be selected and sent to the data services agent in batches, if implemented accordingly. Thus huge amounts of data in the source system will not lead to memory overflows in the source system or the agent.

- No additional datastore for the web service is needed.

Cons:

- The setup of the dataflow is a little more complicated due to the need to store the source code of the form routine.
- Can only can be used to read data from a system
- Has only input parameters, no support for input tables
- Has only one flat output table

For a detailed description of how to set up a Custom ABAP Transform, see Custom ABAP Transform [page 222].

9.12 Consolidation System

There are many use cases in which SAP Integrated Business Planning is used as a central planning tool for several ERP systems.

SAP IBP needs data from other systems, like sales systems or financial systems. At the same time it can also be required to write back the results of the planning to several ERP systems.

You can integrate SAP Integrated Business Planning with several source and target systems using SAP Cloud Integration for data services. In general every task and thus every planned integration job only has one source system and one target system. If many systems are involved a lot of integration jobs might need to be planned. The options to combine data from different systems for one target in SAP IBP are limited and in many cases require the use of file shares for storing intermediate data. This will also increase the number of dataflows and tasks.

For scenarios with several source and target systems it is a good idea to consolidate the data needed for SAP IBP in a central customer system. Many customers use an SAP BW or an SAP APO system for this purpose. This is particularly useful if the data is not aligned between the ERP systems, for example if the IDs are different between the systems or if large parts of the data need to be read from one system, but the IDs to be used in SAP IBP are stored in another system.

10 Tips and Tricks

The following section provides you with best practices and tips to ensure that your integration projects in SAP Cloud Integration for data services run smoothly.

10.1 Integrating SAP IBP with Several On-Premise Systems

The integration content SAP delivers as templates is meant to be used with **one** on premise SAP ERP system (SAP ECC or SAP S/4 HANA) or **one** SAP APO system or **one** SAP Demand Signal Management system. If you want to integrate with several on-premise systems there are some points you should keep in mind:

- If there is data in the on premise systems that is not aligned or likely to fall out of sync in the future, you should align this data in the on-premise system, for example SAP BW, SAP APO, or one of the ERP systems. Neither SAP IBP nor SAP Cloud Integration for data services is a data reconciliation tool. Reconciliation issues should be resolved in the on premise systems.

- Every data replication job in SAP Cloud Integration for data services is based on one or more tasks and/or process definitions. These tasks and process definitions consist of one or more data flows. Every data flow can have at most one source system and one target system. This has the following implications:
  - If you combine data from two source systems (for example SAP ERP and SAP APO) within one data flow, you can make one system the source system and data from the other system can only be read by a web service or an OData service within the same data flow.
  - As an alternative option, you can define a task containing two data flows. The first data flow reads data from one system and stores it in a file share. The second one reads data from the second system, adds data from the file share using a lookup function, and then sends it to the target system.
  - If you need to read similar data from different ERP systems, but with slightly different selection criteria, for example some plants from one ERP system, other plants from the other system, you can do the following:
    - Define a generic datastore called 'ERP', for example, with several configurations (one for every source ERP system, e. g. ERP_100, ERP_200, ERQ_101).
    - Define a task with data flow(s) using datastore 'ERP' as source system.
    - Define and call a replication job consisting of several steps, one for every source ERP system using the different datastore configurations created above.
    - To simplify the selection logic, it might be a good idea to create a customer table or a customer filed in an existing table in the source systems to select the plants per source system, especially if you want to select a large number of plants.

10.2 Integrating SAP IBP with Cloud Systems

SAP Cloud Integration for data services requires the installation of the SAP Data Services Agent.

To connect to other cloud systems, the agent can be installed in an existing on premise server in the landscape. If there is no on premise server in the landscape, you can use a virtual machine in SAP Business Technology Platform. You need a license of SAP BTP to procure such a virtual machine.

The virtual instance must be ordered and paid by the customer. SAP provides the virtual machine and the customer installs and runs the agent on that virtual machine, including monitoring and upgrades of the agent if needed.

10.3 Avoid Mapping Effort by Using Generic Datastores

To avoid the need of adapting the mapping content in SAP Cloud Integration for data services every time you need to change the source or target system, you should use generic datastore IDs, like 'IBP' for the SAP IBP system and 'ERP' for the SAP ERP or SAP S/4 system.

If you have only one test system and one productive system you just can use the default configuration in the sandbox environment for the test system and the default configuration in the production environment for the productive system.

In case you have several development and/or test systems, you can create several configurations for one datastore, for example 'IBP1' and 'IBP2' or 'ER1_100', 'ER1_201', 'ER2_002', etc.. You can then mark one of the datastore configurations as the default configuration. This configuration is used in case no other configuration has been chosen explicitly. On the maintenance screen for datastores there is a second tab where you can define system configurations. A system configuration is a combination of datastore configurations. You could create the system configurations 'IBP1_ER2_002', 'IBP2_ER1_100', for example.

Then you could define tasks and data flows that use the datastores 'IBP' and 'ERP'. When you schedule such a task you can choose the system configuration you want to run the task with. Like that, you can choose the combination of ERP and SAP IBP systems you want the task to be run with. If you do not choose a system configuration or the system configuration does not contain an entry for a certain datastore, the default configuration of that datastore is used.

System configurations can contain more than two datastore configurations.

> **i Note**
>
> You cannot access several configurations of a datastore in one task.

If you need to access two ERP systems within one task, you have to define configurations for two different datastores, for example ERP_EUROPE, ERP_ASIA and ERP_AMERICAS.

10.4 Using a Realistic Test Environment

The test environment should contain data and data volume similar to the productive environment.

The reason is that many mapping steps in SAP Cloud Integration for data services might turn out to be performance-critical and it's not ideal if you only find out about performance issues once you work in the productive environment. In addition, if you choose not to generate coding in the productive system, but to import it from the development or test system, you have additional effort before you can test in the productive system. After every change of the ABAP part of a data flow the generated ABAP coding in the Business Suite system changes, too. Therefore, you have to transport the changed coding by a workbench transport from the development system or test system to the productive system.

Performance issues can also occur when you select data from the SAP IBP system. So it is important to have realistic data there, too.

For additional information on perfomance tuning for reading data from SAP IBP, search for *Best Practices for Extracting Data from SAP IBP* on the Help Portal at http://help.sap.com/ibp.

10.5 Authorizations for the SAP Business Suite RFC user

If you connect SAP Cloud Integration for data services with an SAP Business Suite system, like SAP ERP or SAP APO using an RFC connection, the corresponding RFC user needs a set of authorizations.

The set of authorizations depends on the technologies you need to connect with the system, for example web services, extractors, or table access. For related topics and more details, see the chapter *Configuring SAP Business Suite connectivity* of the SAP Data Services Agent Guide at https://help.sap.com/cpi_ds.

The following table gives an overview of the potentially required authorizations of the RFC user used for the connection from the Data Services Agent to the SAP Business Suite system. For details about the the required attribute settings, see the chapter *SAP user authorizations* of the SAP Data Services Agent Guide.

Authorizations for the SAP Business Suite RFC user

Authorization Object	Usages
Generic Authorizations	
S_RFC	Development, Production: All
	Especially needed to access functions from groups `/BODS/RS_DS_BASIS` and `/BODS/RS_DS_BW`, the technical framework for connecting to the business suite system
S_RFC_ADM	Development, Production: Send data to Data Services Agent using RFC
S_TCODE	Development, Peoduction: All, except web services and OData

Authorization Object	Usages
S_BTCH_ADM	Development, Production: All
	Needed if background processing is activated for datastore
S_BTCH_JOB	Development, Production: All
	Needed if background processing is activated for datastore
S_TRANSPRT	Development:
	Might be needed in the future to generate the report source code in the development system
S_DEVELOP	Development: generate report (Generate and Execute mode), get version, etc.
	Activities 01 (Create or generate), 02 (Change), 03 (Display) needed
	Development, Production: Import and search tables, search columns
	Activity 03 (Display) sufficient in non-DEV system
Extractor-Specific Authorizations (Authorization class RO)	
S_RO_OSOA	Development, Production: Extractors
	Needed to call extractor data sources
S_SDSDEV /	Development, Production: Table based dataflows
S_DSDEV	Used for table search and column search function
S_SDS /	Development: BODS_STYP = 'D'
S_SDSS	Is needed to access the development-specific functions; for more information, see the chapter *Development versus production functions* of the SAP Data Services Agent Guide at https://help.sap.com/cpi_ds)
S_DSAUTH /	Development, Production: All, except web services and OData
S_SDSAUTH	Authorization to run function modules in /BODS/ namespace
S_DSPGMCHK / S_SDSPGMCHK	Production: Execute preloaded
	Authorization to execute pre-loaded report with specified name, this is the only authorization that is needed in addition when switching from Generate and execute to Execute pre-loaded.

Authorization Object	Usages
Application Specific Authorizations (Various Authorization Classes)	
Various	**Extractors**
	Extractors might require additional application specific authorizations dependent on how they are implemented
	Custom ABAP transforms
	Within a custom ABAP transform business logic might be called that requires additional authorizations. It also might be needed to implement a proper authorization check before calling business logic.

For more information on how to find out which authorizations are missing, see Troubleshooting [page 230].

10.6 Overview of Integration Technologies Provided by SAP Cloud Integration for Data Services

SAP Cloud Integration for data services offers several integration technologies for SAP Business Suite:

10.6.1 Transform Type *Source Table*

There are several options for selecting data from a SAP Business Suite system.

Three of them are available using the transform type *Source Table*. As a prerequisite you must add the corresponding table, view, or extractor as a source table to the source datastore of the data flow. The table type *Table* covers tables and views, whereas the table type *Extractors* covers SAP BW extractors.

Tables

There are two ways of accessing tables in the system:

If you use the transform type *Source Table* you can choose which fields from the table you want to read by mapping those fields to fields of the follow-on transform. This transform can be a query, an aggregation, etc.. You define conditions for the selection indirectly by defining filters in the follow-on transform.

Database Views and Projection Views

Instead of using the tables themselves you also can use database views and projection views.

Database views are defined based on one main table and you can optionally join tables by foreign keys. You also can add additional selection conditions for all joined tables. In the field list, you can add only those fields you really need and overwrite the field names. This is especially helpful in cases where the same field name occurs in several joined tables. You can also use this trick to replace the mainly non-speaking German five letter abbreviation table field names in the view with something more readable.

One use case for using database views is to centrally specify selection conditions for tables. This is especially helpful if the database view is used in a lookup function with mode PRE_LOAD_CACHE.

For more information, see the chapter *Lookup Function*.

Another use case is the definition of database views consisting of several tables related to each other by foreign key relations. This helps to reduce the number of source tables that need to be joined together in the SAP Cloud Integration for data services data flows.

Both use cases reduce the complexity of the mapping content in the SAP Cloud Integration for data services data flows.

Additionally, you can limit the list of fields in the database view to those that are needed for mapping in the data flows. This is the only advantage of using projection views. We therefore recommend database views, because they are more flexible.

Extractors

SAP BW extractors can also be used as data sources for data flows. From a data flow perspective they are handled in the same way as tables. This way you can decide which fields to be selected by mapping them to fields of the follow-on transform. Selection conditions can be defined by adding filters to the follow-on transform as well. Be aware that only one condition per field is pushed down to the extractor, but more complex selection criteria are only evaluated after the extractor call. The only exception is a condition with >= for one field followed by a condition <= for the same field, which is pushed down as a between statement if no other conditions exist for the same field.

> **i Note**
>
> Please be aware that for using extractors a minimum SAP NetWeaver release 7.0 SP8 is recommended. This is the minimum release to support operational data provisioning. On lower releases, extractors might not work at all as a data source for SAP IBP integration or have technical restrictions.
>
> For more information on enabling extractors for the ODP framework please have a look at the following SAP Community Network entry: https://wiki.scn.sap.com/wiki/display/BI/ODP+enabled+extractors#ODPenabledextractors-EnableExtractorsforODPframework

10.6.2 Export DataSources not Exposed to ODP

Following the implementation of SAP note 2350464, customer-defined extractors are automatically released for ODP.

Alternatively, you can use the program `RODPS_OS_EXPOSE` to release customer-defined extractors directly in the relevant source system. For more information, see SAP note 1585204.

10.6.3 Lookup Function

If you just need one or two fields from a table added to a transform, you can define a mapping using function `lookup ()`, which does a lookup of a table field.

This is either done in single entry mode or by caching data (single entry or whole table mode). You can use database views or projection views instead of tables here. This might be especially interesting if you do not want to read single entries, but you do not want to select the whole table either. In this case you can define a view of the type *Database View* with selection criteria (in transaction SE11 in the source system) and use this view instead of the whole table.

10.6.4 Custom ABAP Transform

Custom ABAP transforms are customer defined form routines that can be used to fill an output table.

The only input to custom ABAP transforms are parameters that are defined in the custom ABAP transform and filled with values from global variables.

How to Set Up a Custom ABAP Transform?

1. Create a dataflow with a source datastore of type *Business Suite*.
2. Create a custom ABAP transform in the dataflow and connect it either directly to the ABAP endpoint or the follow-on ABAP transform (query or aggregation). It is recommended to directly connect to the ABAP endpoint and to use an output structure of the custom ABAP transform that is as close to the target structure in SAP IBP as possible.
3. In the *Transform Details* go to *ABAP Options* and specify the name of the source code file for the form routine in field *ABAP Language File Name*. It should have suffix .aba and can also contain a full file path, for example `C:\ProgramData\SAP\DataServicesAgent\abap\forms\Z_CUSTOM_ABAP_FORM.aba`.
4. Define the fields of the output structure.
5. Complete the non-ABAP part of the dataflow by mapping the custom ABAP output structure fields to the SAP IBP fields in the Target_Query transform.
6. Create an .aba file containing the to be executed form routine and place it at the specified folder and filename. A file could look like follows and just creates one empty entry for a smoke test:

> **⁝ Sample Code**
>
> ```
> FORM <<<FORMNAME>>>.
>
> APPEND INITIAL LINE to <<<OTAB1>>>.
>
> ENDFORM.
> ```

A more sophisticated file could for example, look as follows:

> **⁝ Sample Code**
>
> ```
> FORM <<<FORMNAME>>>.
>
> DATA lv_packagesize TYPE i VALUE 5000.
>
> DATA lt_t006a TYPE SORTED TABLE OF t006a WITH UNIQUE KEY msehi.
>
> FIELD-SYMBOLS <ls_t006a> TYPE t006a.
>
> FIELD-SYMBOLS <ls_outtab> LIKE LINE OF <<<OTAB1>>>.
>
> IF download = 'S'.
>
> lv_packagesize = p_pkgsz.
>
> ENDIF.
>
> SELECT msehi msehl FROM t006a
>
> INTO CORRESPONDING FIELDS OF TABLE lt_t006a
>
> WHERE spras = $param1.
>
> SELECT mara~matnr AS mat_matnr
>
> meins AS uomid
>
> mtart AS mattypeid
>
> maktx AS prddescr
>
> FROM mara LEFT OUTER JOIN makt
>
> ON mara~matnr = makt~matnr
>
> AND makt~spras = $param1
>
> INTO CORRESPONDING FIELDS OF TABLE <<<OTAB1>>>
>
> PACKAGE SIZE lv_packagesize
>
> WHERE mtart = 'FERT'
>
> OR mtart = 'HALB'
>
> OR mtart = 'ROH'
>
> .
>
> LOOP AT <<<OTAB1>>> ASSIGNING <ls_outtab>.
>
> CASE <ls_outtab>-mattypeid.
> ```

```
WHEN 'FERT'.

<ls_outtab>-mattypeid = 'FINI'.

WHEN 'HALB'.

<ls_outtab>-mattypeid = 'SEMI'.

WHEN 'ROH'.

<ls_outtab>-mattypeid = 'RAW'.

ENDCASE.

READ TABLE lt_t006a ASSIGNING <ls_t006a>

WITH KEY msehi = <ls_outtab>-uomid.

IF sy-subrc = 0.

<ls_outtab>-uomdescr = <ls_t006a>-msehl.

ENDIF.

ENDLOOP.

PERFORM form3.

REFRESH <<<OTAB1>>>.

append_flag = 'A'.

ENDSELECT.

ENDFORM. "FORMNAME
```

This custom ABAP source code must fulfill some prerequisites:

- The output table must contain the fields `mat_matnr`, `uomid`, `mattypeid`, `prddescr`, and `uomdescr` with fitting types.
- At least one parameter must be defined. The first parameter must be the language for the text selections.

There are some general requirements for the .aba file:

- The form name must be <<<`FORMNAME`>>> (capital letters!)
- You can optinally define and use global types and variables and additional form routines after the ENDFORM statement.
- <<<`OTAB1`>>> (capital letters!) must be used at least once. It represents the output table of the form routine.
- `SELECT *` does not work, it is interpreted partly as comment.
- `PERFORM form3`. shall be used to send data in batches of `p_pcksz` entries.
- Lines of the source code must not be longer than 72 characters.

7. Validate the dataflow. If it is fine you are ready to start the dataflow in the mode *Generate and Execute* .
   If the execution fails, it might be due to the following reasons:

   - The RFC user to connect to the SAP Business Suite system has insufficient authorizations. Check transaction `SE54` in the system and switch to the RFC user within the transaction).
   - The .aba file cannot be found or does not fulfill the required format.
   - There are problems with the target datastore, for example, locked files.

If the execution was successful the source code of the generated report is placed on the agent server as another .aba file.

If you also want to run the custom ABAP in *Execute pre-loaded* mode you need to locate the .aba file of the report on the agent and copy the source code to a report with the name specified within the file. For the moment it is stored in the directory of the agent .exe file, for example `C:\Program Files\SAP\DataServicesAgent\bin` for a standard Windows installation. This might be corrected in future.

8. Activate the report.
9. Run the custom ABAP in *Execute pre-loaded* mode. This can be done by either changing the setting in the datastore configuration or by using another datastore configuration with *Execute pre-loaded* mode. There are some additional reasons why this might fail:

   - The report with the name given in the Custom ABAP Transform does not exist, is not active or does not fulfill the technical requirements for being called by SAP Cloud Integration for data services.
   - Additional authorizations might be missing.

   If you need to run the dataflow in *Execute pre-loaded* mode also in other systems, you can transport the report to those systems.

10.6.5 RFC-Enabled Function Modules

SAP Cloud Integration for data services together with the latest version of the Data Services Agent supports direct calls of Remote-Function-Call enabled function modules.

For more information, see.

10.7 ABAP Query/ABAP Aggregation Versus Query/ Aggregation

Queries and aggregations can be added in two different ways, either as ABAP query/ABAP aggregation transform or as query/aggregation.

Please note the following differences and pros and cons of the two types of transforms.

When you add at least one ABAP query or ABAP aggregation transform, an ABAP box is created in the dataflow. This ABAP box represents a report that is generated in the source system and executed there. Everything that is defined inside this box is executed by the generated report in the source system. All other transforms of the dataflow, except for *Source Table*, are executed in the agent. ABAP transforms have the following pros and cons.

Pros:

- Logic and selections happens in the source system and therefore causes less network traffic.
- Fewer resources are needed for the agent, as the main logic happens in the ABAP system.
- The most flexible mapping solution using Custom ABAP transform is available (see below).
- Memory consumption might be smaller due to the usage of nested selects instead of joining fully loaded tables.

Cons:

- Creating, updating, and transporting the report source code means additional effort.
- Many SAP Cloud Integration for data services specific functions for calculations are only available in SAP Cloud Integration for data services transforms.
- More resources are needed on the ABAP system, as the main logic happens in the ABAP system.
- Nested selects might be slower than full table loads if complete tables are joined.

10.8 ABAP Query and Execute Pre-loaded: How to Transport the Generated Report

We recommend that you use the mode *Generate and Execute* in **development** datastores only and to use the mode *Execute Pre-loaded* in **productive** datastores. As a prerequisite, the report must exist in the productive system.

> **i Note**
>
> The dataflow must have at least one ABAP Query or ABAP Aggregation or Custom ABAP transform, otherwise there is no generated report involved in the mapping.

The recommended approach to generate and transport the report source code is as follows:

1. Ensure that the default configuration of the source datastore has the ABAP execution option *Generate and execute* selected.
2. In the default configuration of the source datastore there are attributes that influence the report generation in the development system. They are grouped under *Upload Attributes*. The most important one is the *Development Class* (package). Insert a transportable development package here and save your changes.
3. Go to the *Projects* tab and expand the relevant project, if needed.
4. Mark the relevant task or process and click *Edit*.
5. Mark the relevant dataflow and click ▶ ❯ *Actions* ❯ *Edit* ❯.
6. Double-click on one transform of type *Custom ABAP*, *ABAP Query*, or *ABAP Aggregation* and click *Generate and view ABAP report*.
7. On the *ABAP Options* tab, ensure that field *ABAP Program Name* contains the right name for the report to be generated in the development system.
8. Click *Close* in the lower right corner.
9. Click *Generate and view ABAP report*.
10. Select a fitting agent and if you want the report to be generated in the development system, select the checkbox *Load ABAP program to SAP Application server*. Otherwise the source code is only stored as a file on the agent and displayed.
11. Click *Ok* and the report source code will be displayed.
12. Check the log in the lower part of the popup to make sure that no errors occurred.

For more information, search for *Generate and Load an ABAP Program* on the Help Portal at https://help.sap.com/cpi_ds.

Manual Procedure

If the approach above is not possible, for example because the RFC user does not have the required authorizations, you also can manually create the report in the development system as follows:

1. Find or maintain the report name to be used in the corresponding dataflow definition.
   To do so:
   1. Expand the involved project, if needed.
   2. Mark the involved task and click *Edit* or *View*.
   3. Double click one of the ABAP query or ABAP aggregation or custom ABAP transforms.
   4. Click on the tab *ABAP Options*.
   5. In the field *ABAP Program Name* you find the report name to be used. In edit mode, you can change the report name.
2. Log on to the development system.
3. Start transaction se38 by typing **/nse38** in the *OK* code field and pressing **Enter**.
4. Enter the report name you found before.
5. Fill the *Title* field with a descriptive text
6. Select the type *Executable Program* in the dropdown list.
7. Select a status from the dropdown list, for example *Customer Production Program*.
8. Select an application from the dropdown, if you wish.
9. Click *Save*.
10. In the popup, enter a development package or choose one using the search help and click *Save*.
11. In the next popup either select or create a development request.
12. Connect to the data services agent server to do the following:
    1. Find the file with the report source code on the data services agent server.
       - For a Windows installation, you can find it in folder `%DS_COMMON_DIR%\abap`, where `%DS_COMMON_DIR%` by default has the value `C:\ProgramData\SAP\DataServicesAgent`.
       - For a Linux installation, you can find it under `<install dir>/abap`.
       - Exception: Custom ABAP transform, see above.
    2. Open the file with a text editor and copy the whole content to the clipboard of your local machine (by pressing **Ctrl-a** and **Ctrl-c**).
13. Replace the entire source code with the content of the clipboard by pressing **Ctlr-a** and **Ctrl-v**.
14. If you like to, you can replace the generic report name ZAL_ABAP in the first line of the source code with the real report name.
15. Save and activate the report.
16. Now you can use the report in the development system in mode *Execute pre-loaded*. This can be either done in the development client by changing the datastore configuration from *Generate and Execute* to *Execute pre-loaded* or by an additional datastore configuration for the development client or a separate test client of the same system with the setting *Execute pre-loaded*.
17. To run the task in a separate test system or in the productive system, release the transport and import it into that system.
18. If the selection logic or interfaces within the ABAP box of the dataflow have changed, proceed as follows:
    1. Run the task in mode *Generate and Execute* in the development system.
    2. Find the file with the report source code on the data services agent server (see above).
    3. Open the file in a text editor and copy the whole content to the clipboard of your local machine by pressing **Ctrl-a** and **Ctrl-c**.

4. Log on to the development system and start transaction se38 by typing **/nse38** in the *OK* code field and pressing **Enter**.
5. Fill in the report name and click *Change*.
6. Replace the source code with the content of the clipboard by pressing **Ctrl-a** and **Ctrl-v**.
7. Optionally, you can replace the generic report name ZAL_ABAP with the real name.
8. Save and activate, and choose or create a transport request if needed.
9. Continue as described above for the execution and transport steps.

10.9 Find Fields in the Source or Target Field List for Mapping

When you display or edit data flows in SAP Cloud Integration for data services, you can use the browser search function to search for fields by technical name or description.

This works for all source and target fields contained in the browser page, even in scrollable lists.

10.10 Sort Order of Global Variables

To change the sort order of global variables defined in a task, proceed as follows:

Whenever you a edit a global variable in a task, the varbiable is moved to the first position in the list. To sort the global variables, edit the variables starting with the last one in the list and ending with the first one. You do not actually have to change anything, a dummy change is sufficient. You can open the variable, by clicking on *Edit* in edit mode and then leave the popup by clicking *OK*.

10.11 Batch Command 'REPLACE'

You can set the batch command for post-processing in SAP IBP using global variable $G_BATCH_COMMAND. The following values are supported for this variable:

- 'INSERT_UPDATE' (inserts or updates the selected entries, but does not delete any entries)
- 'DELETE' (deletes the selected entries)
- 'REPLACE' (deletes all entries, and replaces them with the selected ones)

If a task is executed in 'REPLACE' mode, all data of the target table is deleted and replaced with the selected data. This does not lead to the intended behavior in the following cases:

- Data is written to the same SAP IBP table by different tasks

In this case, the first task deletes all entries from the table and inserts the selected ones. The second task deletes all entries again and adds the selected ones from the second task. The entries from the first task are gone afterwards.

- The selection is restricted, for example, by changing global variables.
  For example, if you plan a job using the task SOP_MD_ProductMaster with $G_BATCH_COMMAND equal to 'REPLACE', $G_MATERIAL_TYPE_1 equal to 'ROH', and $G_MATERIAL_TYPE_2… _5 equal to '', the outcome is the following: All products are deleted from SAP IBP and only products with the product type 'ROH' are added again. As you did not select the products with product types 'HALB', 'FERT', 'HIBE', and 'FHMI', they are deleted. Dependent data using the deleted products is also deleted.

If you simply set the global variable $G_BATCH_COMMAND to 'REPLACE', the preload script of the dataflow overwrites it to 'UPDATE'. This helps to avoid serious data loss. If you are confident that you want to replace the existing data in the target table with the selected data, besides changing the global variable, also remove the following section from the preload script on the *Execution Properties* tab:

```
if ($G_BATCH_COMMAND = 'REPLACE')
  begin
#     change 'REPLACE' mode to 'INSERT_UPDATE' mode, because 'REPLACE' has a
severe risk of data loss
        if it is not handled correctly
#     if you plan to enable batch mode 'REPLACE' by modifying the behavior here
please read the documentation
        carefully to avoid data loss when running integration jobs the wrong way
    print('WARNING - Value ' || $G_BATCH_COMMAND || ' of variable
$G_BATCH_COMMAND set by input is ignored,
        because this mode is forbidden by preload script');
    $G_BATCH_COMMAND = 'INSERT_UPDATE';
  end
```

11 Troubleshooting

The following chapter provides you with useful links and explanations in case of errors.

To efficiently troubleshoot an issue you are experiencing with SAP Cloud Integration for data services, you can allow SAP Support to temporarily access your system. For more information, see *Enable Access for SAP Support* in the Help Center for SAP Cloud Integration for data services at Enable Access for SAP Support.

Useful Links

The *Help Center for SAP Cloud Integration for data services* contains some helpful troubleshooting documents:

- Troubleshooting task or process failures
- Troubleshooting during task or process development
- FAQs

The *Data Services Agent Guide* for SAP Cloud Integration for data services also contains some troubleshooting chapters:

- Agent Diagnostics
- Stopping the internal database
- Manually uninstalling the agent
- Configuring SAP Business Suite connectivity: This chapter helps you to analyze and solve communication issues with SAP Business Suite systems.

SAP Knowledge Base Articles and SAP Notes

The component for Knowledge Base Articles (KBAs) and SAP Notes concerning the integration of SAP IBP using SAP Cloud Integration for data services is `SCM-IBP-INT-TS-DS`.

For SAP Cloud Integration for data services, the component is `LOD-HCI-DS`.

The component for the integration post-processing in SAP Integrated Business Planning is `SCM-IBP-INT-TS-POP`. For general topics regarding integration of time-series data, you can use the `SCM-IBP-INT-TS` component.

Tricky Errors

During the configuration of your integration project, you might encounter certain errors. you might encounter one of the following errors:

Datastore cannot be created because of missing IBP Instance

If you want to create a datastore and there is no fitting SAP IBP instance available, create incident under the component `LOD-HCI-DS` providing the following information:

- System details for your SAP IBP instance
- The URL for your SAP Cloud Integration for data services system
- Your SAP Cloud Integration for data services organization name (displayed in the top right of the SAP Cloud Integration for data services interface next to the user name)

RFC CallReceive error <Function /SAPDS/RFC_ABAP_INSTALL_RUN: RFC_ABAP_MESSAGE, You do not have authorization for this function ... [4103]>

Add the required authorizations to the RFC user in the corresponding SAP Business Suite system. See Authorizations for the SAP Business Suite RFC user [page 218] for more details. If the list above does not help or is incomplete you can find the missing authorizations in the SAP Business Suite system. To do so, you one of the following transactions:

- Transaction SU53: Display Authorization Data
  Call transaction SU53 in the system you tried to connect from SAP Cloud Integration for data services. It is showing the last failed authorization checks of the logon user. If the RFC user is a different one you can switch to that user by F5 and choosing the name of the RFC user.
- Transaction ST01: System Trace
  Call transaction ST01 to activate a trace for authorization checks by marking the corresponding checkbox. By setting either of the radio buttons All or Error only you can decide if all checks are traced or only the failing ones. In addition you can set a filter for the RFC user name when clicking on button General Filters. Afterwards you can activate the trace, run the replication job in SAP Cloud Integration for data services, switch the trace off again and do an analysis of the trace.

records failed with error, special characters are not allowed

If a task loaded to SAP Integrated Business Planning fails with the following error message: " # records failed with error, special characters are not allowed", you can use an SAP Cloud Integration for data services function to remove the special characters. For more information, see SAP note 2007254 .

<task name> is currently in use and locked by <user name>. Try again later.

If you are sure that the user is not working on the task any more you can unlock the task by selecting it and clicking ▌▶ > *More Actions* > *Unlock* ▌.

Entries are missing in transaction LBWE preventing extractor activation

You have activated extractors, for example `2LIS_12_VCITM` in transaction `RSA5`. However, the extractor is not visible in transaction `LBWE`. This is probably due to a transport issue for entries in tables `TMCEXACT` and `TMCEXUPD`. These entries should be available in client 000 (or others) and can be copied from there.

Attribute values having only SPACE characters are not allowed

When using SAP Cloud Integration for data services, you can solve the issue as follows if the field shall not have any value.

In the mapping of the corresponding field instead of <source_field>, use the following expression:

```
ifthenelse(ltrim_blanks(<source_field>) = '', NULL, <source_field>)
```

This way all empty strings and spaces are replaced by NULL values, which will be accepted by the SAP IBP system. However, if the field is mandatory still another error message might occur.

The combination with replacement of non-supported special characters could look like this:

```
ifthenelse(ltrim_blanks(translate(<source_field>, CHR(10) || CHR(13) || CHR(39) ||
'"<>', ' ')) = '', NULL, translate(<source_field>, CHR(10) || CHR(13) || CHR(39) ||
'"<>', ' '))
```

The easiest way to solve this issue is to make the target SAP IBP system convert fields only containing spaces into NULL values before processing the master data import. This system-level setting can be switched on by configuring the following global parameter (this setting is off by default):

PARAMGRP: INTEGRATION

PARAM: SPACE_TO_NULL

PARAMVAL: **1**

For more information and alternative solutions, see SAP Note 2435673 .

CI-DS task fails due to the not nullable BATCH_TIMESTAMP column

The `$G_LOAD_DATE` variable is translated to `BATCH_TIMESTAMP` in SAP IBP, and if it is left empty, this error occurs. You can resolve this issue by setting the `$G_LOAD_DATE` variable at runtime or in the *Execution Properties* of the task. We recommend to use the `sysutcdate()` function as a value for this variable. For more information, see SAP Note 2758252 .

Submitted data has older timestamp <datetime in UTC> than what is already available in the system <datetime in UTC>

For more information on how to deal with this error, see 2105719 .

CI-DS and IBP add timestamps to tasks in different time zones

When data is transferred from SAP Cloud Integration for data services to SAP IBP, you might find that the two systems assign timestamps to tasks according to different time zones. To resolve this, set the global variable `$G_LOAD_DATE = SYSUTCDATE()` in the *Execution Properties* of the SAP Cloud Integration for data services task and run the task again. For more information, see SAP Note 2483783 .

Error: <"curl_easy_perform":56:Failure when receiving data from the peer> with small recordset

This issue is often resolved after re-registering the SAP Data Services Agent. For more information, see SAP Note 2639185 .

Post Processing Error Handling

SAP Note 2436131 (*IBP: Common post processing jobs errors in HCI*) is handling the following error messages:

- XXXX records failed with error, PERIODID is missing for the KEYFIGUREDATE provided caused by: Missing period ID.
- XXXX records failed with error, Unable to import the data because the row is duplicated in batch &1 and file &2 caused by: Key figure duplication.
- XXXX records failed with error, Attribute value combination does not exist: PLAREA &1 PLEVEL &2 PLSET &3 caused by: Attribute value combination does not exist.
- XXXX records failed with error, Master data does not exist for root attribute &2 in source master data type &1 for planning level &3 caused by: Master data not available.
- XXXX records failed with error, List of attributes that exist in the row does not match the list of root level attributes for certain key figure(s) base level in batch &1 and file &2 caused by: Key figure attributes do not match root level.

Data integration report cannot be downloaded in SAP IBP

In case you have too much data to download, use the reporting tables in the SAP Cloud Integration for data services system. For more information, search for *Viewing Import Job Status* on the Help Portal at https://help.sap.com/ibp.

Key figure upload to version fails with fatal error and exception
`CX_AMDP_EXECUTION_FAILED`

In case the exception `CX_AMDP_EXECUTION_FAILED` is shown when integrating key figure values to a version other than the baseline, make sure that the key figure used in the task is selected in the appropriate version. If you want to upload the key figure to the version with a different value than in the baseline version, select *Version-Specific Key Figure* under *Versions* in the *Planning Areas* app.

11.1 Pushdown of Filter Conditions To Extractors

Many extractors and data sources support the definition of filter criteria to shrink the amount of selected data in case not all data are needed in the target system.

In the case of extractors in the IBP add-on for SAP S/4HANA and SAP ERP, the pushdown of the filters is important for the following reasons:

- Pushed down filters reduce the amount of selected data in the source system and the amount of data sent to SAP Cloud Integration for data services and thus improves the performance.
- Some of the filter fields, especially `TIMEPROFILEID` and `TIMEPROFILELEVEL` directly influence the way key figure data is aggregated in the source system.
- If a filter of a key figure extractor is not pushed down to the extractor, it needs to be requested from the extractor, although it is not necessary for mapping a target table field, but only needed to evaluate it in the dataflow after the extractor call. This can lead to duplicate entries for the same key combination if the aggregation is not repeated in the dataflow.
  Therefore, it is important to make sure that the filter conditions are pushed down to the extractors.
  Supported filter conditions are defined on field level of the extractors. You can use transaction RSA2 in the source system to get an overview on fields support filter criteria as follows:
  1. Call the RSA2 transaction.
  2. Select the extractor.

3. Choose *Display* and navigate to the tab fields.

Column Selection Poss. (SELSTATE) shows if the field supports filter conditions, and the field *Sel. Options (SELOPTS)* describes what sorts of filters are supported, and if its value is zero, EQ (Equals to) and BT (Between) are supported. Otherwise, the following value list applies:

Supported Filters

Value	Description	Binary Value
EQ	Equal To	1
BT	Between	2
CP	Pattern	4
GT	Greater Than	8
GE	Greater Than or Equal To	16
LT	Less Than	32
LE	Less Than or Equal To	64
NE	Not Equal To	128
NB	Not Between	256
NP	Exclude Pattern	512

If the value of the field *Sel. Options (SELOPTS)* is the sum of some of the above values, all of them are valid. For example, value 3 allows EQ and BT, and value 1023 allows all types of filters.

Filter conditions of extractors are defined in a table parameter, which has the following columns:

Filter Conditions of Extractors

Field Name	Description
FIELDNM	Name of the extractor field
SIGN	Indicator if the entry is I (Including) or E (Excluding)
OPTION	Defining the type of comparison, possible values are listed in the column *Value* of the table above.
LOW	Value, lower limit or pattern to compare with
HIGH	Upper limit for the comparison for options BT and NB

> **i Note**
>
> Define the filters for extractors in the first transform after the extractor definition. In case they are defined in later transforms, the risk that they are not pushed down increases.

Filter conditions defined for extractor fields in SAP Cloud Integration for data services only support the filter types EQ and BT as shown in the examples below:

FIELD1 = ' **A**' and FIELD2 = ' **B**' is pushed down as

> **⇶ Sample Code**
>
> ```
> FIELD1 I EQ A
> FIELD2 I EQ B
> ```

FIELD1 =' **A**' or FIELD2 =' **B**' is not pushed down. Conditions for different fields always need to be combined with and

FIELD1 >=' **C**' and FIELD1 <= ' **E**' is pushed down as

> ⇘ Sample Code

```
FIELD1 I BT C E
```

FIELD1 >=' **C**' is not pushed down. As a workaround, you can define FIELD1 >= '**C**' and FIELD1 <=' **ZZZZZ**', which will be pushed down as

> ⇘ Sample Code

```
FIELD1 I BT C ZZZZZ
```

FIELD1 <' **C**' is not pushed down. As a workaround, you can define FIELD1 >=' **A**' and FIELD1 <=' **BZZZZZ**', which will be pushed down as

> ⇘ Sample Code

```
FIELD1 I BT A BZZZZZ
```

FIELD1 LIKE ' **ABC%**' is not pushed down. As a workaround, you can define FIELD1 >=' **ABC**' and FIELD1 <= '**ABCZZZZZ**', which will be pushed down as

> ⇘ Sample Code

```
FIELD1 I BT ABC ABCZZZZZ
```

FIELD1 LIKE ' **%DEF**' is not pushed down, and there is no workaround.

FIELD1 = ' **A**' or FIELD1 = '**C**' is pushed down as

> ⇘ Sample Code

```
FIELD1 I EQ A
FIELD1 I EQ C
```

FIELD1 = ' **A**' and (FIELD2 = ' **C**' or FIELD2 = ' **E**') is pushed down as

> ⇘ Sample Code

```
FIELD1 I EQ A
FIELD2 I EQ C
FIELD2 I EQ E
```

FIELD1 = ' **A**' and (FIELD2 = ' **C**' or FIELD2 = ' **E**') and (FIELD3 = ' **G**' or FIELD3 =' **J**') is pushed down as

> ⇘ Sample Code

```
FIELD1 I EQ A
```

```
FIELD2 I EQ C
FIELD2 I EQ E
```

That is, the push-down is incomplete and leads to potential issues.

To check which filter conditions are pushed down to the extractor, the easiest way is to call transaction ODQMON, change the field *Request Select.* to *Without Subscription (Full)* and navigate to the correct extractor by double-clicking the corresponding *Requests* field. The pushed-down selection conditions can be seen in column *Selection* at the end of the list. In the case of earlier NetWeaver releases, this column is not supported.

12 Related Documentation

Documentation	Location
Help Portal for SAP Integrated Business Planning	http://help.sap.com/ibp
Help Portal for SAP Cloud Integration for data services	https://help.sap.com/viewer/product/SAP_CLOUD_PLAT-FORM_INTEGRATION_FOR_DATA_SERVICES
Data Export Guide for SAP Integrated Business Planning for Supply Chain	Search for *Data Export from SAP Integrated Business Planning* on the Help Portal at http://help.sap.com/ibp
SAP Data Services Agent Guide	SAP Data Services Agent Guide
Creating and Configuring Service Providers and Service Consumers in SAP NetWeaver Library	▌ *SAP NetWeaver Library* ❯ *Function-Oriented View* ❯ for your SAP NetWeaver release at http://help.sap.com/nw

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.

THE BEST RUN SAP