Integration Guide | PUBLIC
Document Version: 1.1 – 2024-05-15

Integrating Product Data from SAP S/4HANA Cloud to SAP Integrated Business Planning

Integrating SAP IBP for Supply Chain 2402 with SAP S/4HANA Cloud Using SAP Cloud Integration

THE BEST RUN SAP

Content

1 Introduction. 4

2 Prerequisites. 5

3 Data Collection. 6

4 Configuring the Integration Flow. 9

4.1 Configuring the Authentication. 9

4.2 Data Mapping. 10

4.3 Defining Additional Parameters. 11

4.4 Working with Field Extensions. 14

4.5 Working with the Expand Field. 16

4.6 Working with the Filtering Fields. 17

4.7 Scheduling the Integration Flow. 18

5 Troubleshooting. 19

Document History

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

Version	Date	Description
1.1	May 15, 2024	Link for `Product` API has been updated. Parameters inDefining Additional Parameters [page 11]section have been updated.Working with the Expand Field [page 16] and Working with the Filtering Fields [page 17] sections have been added.
1.0	February 16, 2024	Initial version

1 Introduction

Using the integration flow, you can integrate data from SAP S/4HANA Cloud, as part of the solution SAP S/4HANA Cloud Public Edition to SAP Integrated Business Planning for Supply Chain (SAP IBP). Using this data, you can perform demand forecasting in SAP IBP, then integrate the results back to SAP S/4HANA Cloud as planned independent requirements.

Data integration between SAP IBP and SAP S/4HANA Cloud using the integration flows in the *SAP IBP - Integration with SAP S/4HANA Cloud* package is available with SAP IBP 2402 and higher.

The *Integrate Products from SAP S/4HANA Cloud to SAP IBP* integration flow collects master data from SAP S/4HANA Cloud and transfers them into the corresponding master data types in SAP IBP. Data is integrated from the following master data types:

- *Product* (`PRODUCT`)
- *Location Product* (stored only on the platform for filtering)
- *Unit of Measure* (`UOMTO`)
- *Unit of Measure Conversion Factor* (`UOMCONVERSIONFACTOR`)

The listed master data types are also the source master data types of the base planning level of the `ACTUALSQTY` key figure, which can be integrated using the *Integrate Sales Order History Data from SAP S/4HANA Cloud to SAP IBP* integration flow.

The *Integrate Products from SAP S/4HANA Cloud to SAP IBP* integration flow receives data through the `Product` OData service. For more information, see Product (Version 2).

2 Prerequisites

We recommend that you have configured frequently used parameters using the *Define Default Values for Data Integration Between SAP IBP and SAP S/4HANA Cloud* integration flow.

Note that you can also define the ID and description of the base unit of measure with the *Define Default Values for Data Integration Between SAP IBP and SAP S/4HANA Cloud* integration flow.

3 Data Collection

Data is collected in SAP S/4HANA Cloud via OData calls using the following requests:

Request 1: Product

This request collects data from the `A_Product` OData entity. For more information about this entity, see https://help.sap.com/docs/SAP_S4HANA_CLOUD/3c916ef10fc240c9afc594b346ffaf77/2c973258fb9d2060e10000000a44147b.html. The request selects the following fields:

- `Product`
- `ProductType`
- `CreationDate`
- `LastChangeDate`
- `CrossPlantStatus`
- `BaseUnit`

You can override which fields are selected using the `Product Fields` parameter.

Data Filtering

Product range filtering is collected from the `Product Filter` parameter.

Additional filters are applied first based on the `Further Filters for Products` parameter, then based on the `Further Filters for Plants` parameter.

The list of plants defined in the `Plant Filter` parameter of the *Define Default Values for Data Integration Between SAP IBP and SAP S/4HANA Cloud* integration flow provides additional filtering through the `ProductPlantSupplyPlanning` entity.

The `PlanningStrategyGroup` entity further narrows down the results through the `ProductPlant` entity navigation link. Planning strategy groups are collected prior to the first request in a separate OData call. You can view the list of the collected planning strategy groups in the *Manage Variables* app using the `PlanningStrategyGroups` ID. If planning strategy group data changes in SAP S/4HANA Cloud, delete the variables before running the *Integrate Products from SAP S/4HANA Cloud to SAP IBP* integration flow again.

Request 2: Product Plant Supply Planning

This request selects the following fields:

- `Product`
- `Plant`

- `MRPResponsible`
- `SafetyStockQuantity`

You can override which fields are selected using the `Product Fields` parameter.

Data Filtering

The same filters are applied as on product data, however, there is no navigation required since all parameters are available in the entity.

Request 3: Product Unit of Measure

This request collects data related to product unit of measure. It selects the following fields:

- `Product`
- `AlternativeUnit`
- `QuantityDenominator`
- `QuantityNumerator`
- `BaseUnit`

You can override which fields are selected using the `Unit of Measure Fields` parameter.

Data Filtering

The same filters are applied as on product data, however, there is no navigation required since all parameters are available in the entity.

Request 4: Product Description

This request collects data related to product descriptions.

Data Filtering

Data is filtered based on the language (English by default). You can change it using the `Language For Descriptions` parameter.

Other data filters are applied like in the case of product data.

Request 5: Product Plant

This request collects data related to product and plant combinations. It selects the following fields:

- `Product`
- `Plant`

The product and plant combinations are stored in a shared datastore as `ProductPlantFilters`. The list of combinations are refreshed with every run of the integration flow. The ID of the stored list of products can be changed in the `Datastore ID for Product Plant Filter` parameter.

Data Filtering

The same filters are applied as on product data, however, there is no navigation required since all parameters are available in the entity.

4 Configuring the Integration Flow

4.1 Configuring the Authentication

The integration flow connects to both the SAP S/4HANA Cloud and the SAP IBP system. Connections, including the authentication method, must be created and configured at different places depending on the respective system. Once the connections are created for both directions, you need to configure them in the integration flow under *Configure* > *Receiver*.

Authentication Methods for the Connection to SAP IBP

You can only choose basic authentication when connecting to SAP IBP. You can configure the authentication method during the configuration of the destination. You can set the name of the destination using the `Destination for SAP IBP` parameter of the integration flow.

For more information, see Setting Up the Integration.

Authentication Methods for the Connection to SAP S/4HANA Cloud

The following authentication methods are available when connecting to SAP S/4HANA Cloud:

- Basic authentication
- Client certificate (X.509 certificate)

You can select the authentication method in the integration flow under *Configure* > *Receiver* > *Authentication*. Although there are more options displayed in the list, only basic authentication and client certificate authentication are supported.

The default authentication method is client certificate.

Setting Up the Client Certificate Authentication Method

As a prerequisite of using a client certificate, add and deploy the required key pair to the keystore. You can do so in SAP Integration Suite using the *Keystore* tile in the *Manage Security* section under *Monitoring Artifacts*. For more information, see https://help.sap.com/docs/cloud-integration/sap-cloud-integration/managing-keystore-entries.

If you select authentication using a client certificate when configuring the integration flow, you need to enter the private key alias.

Setting Up the Basic Authentication Method

As a prerequisite of using basic authentication, create and deploy the user credentials type of security material. You can do so in SAP Integration Suite using the *Security Material* tile in the *Manage Security* section under *Monitoring Artifacts*. For more information, see https://help.sap.com/docs/cloud-integration/sap-cloud-integration/managing-security-material.

If you select basic authentication when configuring the integration flow, you need to enter the credential name.

4.2 Data Mapping

The following default data mapping is available in the integration flow:

Mapping of Product Data

The product master data type in SAP IBP is mapped to the `Product` OData entity in the following way:

Property in OData Entity	Field in SAP IBP	Further Hints
`Product`	`PRDID`	
`ProductDescription`	`PRDDESCR`	The property is available from the `Product` entity through the *ProductDescription* > *ProductDescription_Type* > *ProductDescription* navigation.
`ProductGroup`	`PRDGROUP`	
`BaseUnit`	`UOMID`	
The description of `BaseUnit`	`UOMDESCR`	

Mapping of Unit of Measure Data

The unit of measure master data type in SAP IBP is mapped to the collection from the `Product` OData entity.

The distinct values of each `BaseUnit` and `AlternativeUnit` properties of products are taken using the *Product* > *Product_Type* > *BaseUnit* and *Product* > *Product_Type* > *ProductUnitOfMeasure* > *ProductUnitOfMeasure_Type* > *AlternativeUnit* navigations, and mapped to the `UOMTOID` master data type.

The unit of measure descriptions is collected from the `Texts` OData entity service. The `UOMDESCR` attribute is mapped to the `Name` property.

Mapping of Unit of Measure Conversion Factor Data

Conversion factor data is available through the *Product* > *Product_Type* > *ProductUnitOfMeasure* > *ProductUnitOfMeasure_Type* navigation of the `Product` OData entity. The unit of measure conversion factor master data type in SAP IBP is mapped to the `ProductUnitOfMeasure_Type` OData entity in the following way:

Property in OData Entity	Field in SAP IBP	Further Hints
`Product`	`PRDID`	
`AlternativeUnit`	`UOMTOID`	
`QuantityDenominator`	`UOMCONVERSIONFACTOR`	The `UOMCONVERSIONFACTOR` attribute is mapped to `QuantityDenominator` divided by `QuantityNumerator` calculated to six decimal places.

4.3 Defining Additional Parameters

Under *Configure* > *More*, you can find the following parameters that you can use to configure the integration flow:

Parameter Name	Default Value	How to Configure the Parameter?
`Batch Name`	`Product Run ID: ${header.SAP_MplCorrelationId}`	Define the name of the data batch. This name also identifies the corresponding job in the *Data Integration Jobs* app.
`Datastore ID for Product Plant Filter`	`-keep default-`	Optionally, define the datastore to be used for further filtering data. This filtering ensures that only existing product plant combinations are integrated.
`Destination for SAP IBP`	`-keep default-`	Enter the name of the SAP IBP system to which product data is transferred.
`Field Extensions for Products`		Optionally, you can define complex mappings for certain columns.
`Field Extensions for Unit of Measures`		Optionally, you can define complex mappings for certain columns.

Parameter Name	Default Value	How to Configure the Parameter?
Field Extensions for UoM Conversion Factors		Optionally, you can define complex mappings for certain columns.
Further Filters for Plants		Optionally, define additional filters for the request for location extraction.
Further Filters for Products		Optionally, define OData filters besides the product range to be applied when transferring the products read through the OData service.
Host for SAP S/4HANA Cloud	-keep default-	Define the base URL of the SAP S/4HANA Cloud API
Language for Descriptions	EN	Enter the code of the language that you want to be used of the descriptions of units of measure.
Master Data Prefix	-keep default-	Optionally, define a three-character-long prefix to be used in SAP IBP.
Master Data Types	PRODUCT,UOMTO,UOMCONVERSIONFACTOR	Enter the technical names of the master data types in SAP IBP that you want to integrate.
Parallel Processes	5	Optionally, define the number of threads to be used to read from the OData service.
Planning Area	-keep default-	Define the planning area in SAP IBP to which you want data to be integrated.
Planning Area Version	-keep default-	Define the version of the target planning area in SAP IBP.
Process Unchanged Data	-keep default-	If you want to skip unchanged data, regardless of SAP IBP global parameter setting for this behavior, set this parameter to **false**. If you want to process unchanged data, regardless of SAP IBP global parameter setting, set this parameter to **true**. If you leave this parameter field empty, SAP IBP global parameter will determine the behavior.

Parameter Name	Default Value	How to Configure the Parameter?
`Product Attributes in SAP IBP`	`PRDID,PRDDESCR,UOMID,UOMDESCR`	Define the fields of the `Product` master data in SAP IBP into which you want to integrate data.
`Product Fields`	`Product,ProductType,CreationDate,LastChangeDate,CrossPlantStatus,BaseUnit`	Define the properties of products that you want to be read from the OData service.
`Product Filter`	`-keep default-`	There is a simplified filter to narrow down the list of master data you want to integrate. Every downloaded master data is ordered by the `Product` property. Optionally, you can define a range of products to narrow down the data load.
`Product Plants Per Package`	`5000`	Optionally, define the number of product plants to be included in a data package.
`Unified Base Unit`	`-keep default-`	Define the base unit of measure.
`Unified Base Unit Description`	`-keep default-`	Enter the description of the base unit of measure.
`Expand for Products`		Enter a navigation link for the `Product` collection. For further information, see Working with the Expand Field [page 16].
`Unit of Measure Attributes in SAP IBP`	`UOMTOID,UOMTODESCR`	Define the fields of the `Unit of Measure` master data in SAP IBP into which you want to integrate data.
`Unit of Measure Fields`	`Product,Plant,MRPType,MRPResponsible,SafetyStockQuantity`	Define the properties of the unit of measure that you want to be read from the OData service.
`UoM Conversion Factor Attributes in SAP IBP`	`PRDID,UOMTOID,UOMCONVERSIONFACTOR`	Define the fields of the `Unit of Measure Conversion Factor` master data in SAP IBP into which you want to integrate data.

To use the values defined in the *Define Default Values for Data Integration Between SAP IBP and SAP S/4HANA Cloud* integration flow, use the `-keep default-` value for the relevant parameters. This is also the default value of all parameters for which you can maintain a reusable default value in the *Define Default Values for Data Integration Between SAP IBP and SAP S/4HANA Cloud* integration flow.

4.4 Working with Field Extensions

With field extensions, you can specify additional attributes to integrate data from and change data mapping.

In general, the required syntax of the value of the `Field Extensions` parameter is the following:

Sample Code

```
<FIELDNAME value = 'DESIRED VALUE' skip="DESIRED VALUE" nil = "DESIRED
VALUE">
```

Note

The `FIELDNAME` must be a field that is listed in the `Attributes in SAP IBP` parameter.

The `DESIRED VALUE` can be defined as a constant value, such as `"0"` or `"TEXT"`. If you use a constant value, all the rows are filled with this value for the given field. The entered values of `skip` and `nil` are evaluated as either true or false. The value entered after `skip` is skipped, and the value entered after `nil` is nullified. Note that using the `skip` and `nil` parameters is optional, and that instead of skipping a constant value, you can skip mapping itself.

You can also define the `DESIRED VALUE` as a function mixed with an XPath expression. This way, you can select specific values from the data set or define a logic using exact values.

Example

Using the following code, you can define `CUSTOMFIELD` to be `CustomfieldXXX` where `XXX` is the ID of the corresponding row in the data set:

```
<CUSTOMFIELD value = "concat('Customfield',./ID)">
```

Example

Using the following code, you can skip the field for a certain ID value:

```
<CUSTOMFIELD value = "./DESIREDS4FIELD" skip = "ID='ID value'">
```

You can define an evaluation like the above for any of the fields and with different logical functions. Operations such as `FIELD != ''` also work.

Note

The value of the `DESIREDS4FIELD` can be any of the fields that are requested from SAP S/4HANA Cloud. In the CDS view, you can check which fields are included in the request. You cannot extend the list of the fields in the request, however, you can cycle through the values of the data set using an XPath expression.

In general, the data structure of an XPath expression looks as follows:

Sample Code

```
<item>
```

```
<field1>value1a</field1>
<field2>value1b</field2>
</item>
<item>
<field1>value2a</field1>
<field2>value2b</field2>
</item>
…
```

Based on the above sample, to select `value1a` and `value2a`, use `./field1`, and to select `value1b` and `value2b`, use `./field2`.

Note that the structure of the data can be different at this stage, therefore, it is recommended to always check the structure of the data set before executing the XPath selection.

> **Note**
>
> Although the integration flow validates the syntax of the field extension XML, you need to make sure that its content is defined according to your business needs.

Extending Product Data

There are the following sets of parameter groups for data extension:

- Product
  - `Product Fields`
  - `Product Attributes in SAP IBP`
  - `Field Extensions for Products`
- Unit of measure
  - `Unit of Measure Attributes in SAP IBP`
  - `Unit of Measure Fields` (also used by the unit of measure conversion factor parameter group)
  - `Field Extensions for Unit of Measures`
- Unit of measure conversion factor
  - `UoM Conversion Factor Attributes in SAP IBP`
  - `Unit of Measure Fields` (also used by the unit of measure parameter group)
  - `Field Extensions for UoM Conversion Factors`

All sets of parameters work in the same way. The following description of product-related extensions can be applied to any of the parameter sets.

The `Product Fields` parameter defines the properties in the `Product` OData service to be selected for further use.

The `Product Attributes in SAP IBP` parameter defines the target attributes to be loaded with data in SAP IBP. It also serves as a validation list for the `Field Extensions for Products` parameter, as only attributes listed in the `Product Attributes in SAP IBP` parameter are accepted as attributes for field extension.

You can enter custom extensions in the `Field Extensions for Products` parameter such as the following:

> ### Example
>
> `Field Extensions for Products`:
>
> `<PRDDESCR value="'constant example string'" skip="Product='TEST'" />`
>
> `Product Fields`:
>
> `PRDID,`**`PRDDESCR`**`,UOMID,UOMDESCR`
>
> `Product Attributes in SAP IBP`:
>
> `Product,ProductType,CreationDate,LastChangeDate,CrossPlantStatus,BaseUnit`

The value property defines what fills that attribute. In the previous example, it's a constant value.

The `skip` property requires a Boolean evaluation. If the evaluation is true, then the property is skipped in the given row. In the previous example, if the `Product` key is `TEST`, then the product description property is skipped.

The `nil` property works the same way as the `skip` property.

> ### Example
>
> In the following example, an extra attribute is added:
>
> `Field Extensions for Products`:
>
> `<LEGACYPRODUCTID value="./ProductOldID" />`
>
> `Product Fields`:
>
> `PRDID,PRDDESCR,UOMID,UOMDESCR,`**`LEGACYPRODUCTID`**
>
> `Product Attributes in SAP IBP`:
>
> `Product,ProductType,CreationDate,LastChangeDate,CrossPlantStatus,BaseUnit,`**`ProductOldID`**
>
> This way old product IDs can be stored, however, note that in this example, the product master data had to be changed previously in order to have an attribute like `LEGACYPRODUCTID`.

4.5 Working with the Expand Field

You can only perform a selection of properties of the subnodes using an OData expand operation to retrieve properties for links, such as the `_ProductPlant/PeriodType` property.

> ### Note
>
> Filtering doesn't require the expand operation, only selection does.

Example

Field Extensions for Products: `<EXAMPLEATTRIBUTE value="distinct-values(./_ProductPlant/ProductPlant_Type/PeriodType)"/>`

Expand for Products: `_ProductPlant`

To insert the correct path for the `./_ProductPlant/ProductPlant_Type/PeriodType` value part, you can load the OData service metadata. Based on the metadata, you can determine the appropriate route for the value as the following:

`./_expandlink/enitity_type/property.`

Alternatively, you can enable tracing for the **Integrate Products from SAP S/4HANA Cloud to SAP IBP** integration flow and use the body data of the *XSLT Map S4 to IBP* step as a reference. In this particular instance, the initial portion of the data appears as follows:

Sample Code

```
<?xml version="1.0" encoding="utf-8"?>
<Product xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Product_Type>
<CreationDate xmlns:multimap="http://sap.com/xi/XI/
SplitAndMerge">2019-10-17</CreationDate>
<CrossPlantStatus xmlns:multimap="http://sap.com/xi/XI/SplitAndMerge"/>
<LastChangeDate xmlns:multimap="http://sap.com/xi/XI/
SplitAndMerge">2022-09-05</LastChangeDate>
<_ProductPlant xmlns:multimap="http://sap.com/xi/XI/SplitAndMerge">
<ProductPlant_Type>
<PeriodType>W</PeriodType>
```

Note

The expand operation doesn't include any filtering mechanism, which can result in loading all attributes of the referenced entity. This process can significantly impact performance, so we recommend that you apply filtering at the product level. If there are difficulties with the OData load caused by the large amount of data, you can reduce the package size using the `Product Plants Per Package` parameter.

4.6 Working with the Filtering Fields

To load the necessary products, you can use various filtering inputs.

In the *Product Filter* and *Plant Filter* input fields, you're allowed to list elements, which are separated by commas, or you can define a range, using a hyphen. It's also possible to mix both methods together. The OData filters generated from these input fields are combined using the `AND` operator, which narrows down the results even further.

Example

Product Filter: **`Prod1, Prod3-Prod10`**

Plant Filter: **`Plant1, Plant3`**

> Equals to the OData filter: `(Product eq 'Prod1' or (Product ge 'Prod3' and Product le 'Prod10')) and (Plant eq 'Plant1' or Plant eq ' Plant3')`

The *Further Filters for Plants* and the *Further Filters for Products* field work by adding additional OData filters to the product and plant filter fields. Data is collected from collections such as `Product`, `ProductDescription`, `ProductUnitOfMeasure`, or the `ProductPlant`.

For instance, when you filter for the `BaseUnit` property using the `Product` collection, navigation isn't required. However, when you filter for `BaseUnit` using the `ProductDescription` collection, you need to use the `_Product/BaseUnit` navigation.

To support this process, we provide pre-defined placeholders that you can insert before the attributes. The integration flow replaces these placeholders with the appropriate navigation based on the request.

Use `(pr/)` for the `Product` and `(sp/)` for the `ProductPlantSupplyPlanning` collections as follows:

> ✤ Example
>
> *Further Filters for Products:* **`pr/BaseUnit eq 'ST'`**
>
> *Further Filters for Plants:* **`sp/MRPType eq 'PD'`**

4.7 Scheduling the Integration Flow

You can schedule the execution of the integration flow under *Configure* > *Timer*.

By default, the start of the integration is scheduled for 2100-01-01 to prevent unnecessary integration jobs during the initial deployment. After you've finalized the configuration of the integration flow, you can manually set the timer according to your needs.

You can select *Run Once* to start integration directly. You can also schedule the job for a future date or make it recurring. For more information about scheduling, see https://help.sap.com/docs/cloud-integration/sap-cloud-integration/define-timer-start-event.

5 Troubleshooting

In case an error occurs during integration, you can check the following attachments in the *Monitor Message Processing* view:

- *Processing Results*
  In the *Processing Results* attachment, you can find a summary of the status of post-processing in SAP IBP.
- *Parameters*
  In the *Parameters* attachment, you can find all the parameters of the *Integrate Products from SAP S/4HANA Cloud to SAP IBP* integration flow together with the parameters of the *Define Default Values for Data Integration Between SAP IBP and SAP S/4HANA Cloud* integration flow. In this attachment, you can check the values of the parameters that were used when creating the query and scheduling the upload after the query.
- *Error Messages*
  In the *Error Messages* attachment, you can find all types of error messages coming from SAP IBP. This attachment is only available if errors occurred.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

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

www.sap.com/contactsap

THE BEST RUN SAP