Integration Guide | PUBLIC
Document Version: 1.0 – 2024-11-25

SAP IBP - Reusable Integration Flows Examples

Data Integration Examples Between SAP IBP for Supply Chain 2411 and Higher and SAP Cloud Integration Using SAP IBP - Reusable Integration Flows

THE BEST RUN SAP

Content

1 Document History. 3

2 Introduction. 4

3 Prerequisites. 5

4 SAP IBP Read - Example. 6

4.1 Introduction to SAP IBP Read - Example. 6

4.2 Using SAP IBP Read - Examples. 6

4.3 Reading Various Data Types. 8

5 SAP IBP Write - Example. 10

5.1 Introduction to SAP IBP Write - Example. 10

5.2 Using SAP IBP Write - Examples. 10

5.3 Writing Various Data Types. 12

1 Document History

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

Version	Date	Description
1.0	November 25, 2024	Initial version

2 Introduction

This document describes the *SAP IBP - Reusable Integration Flows Examples* demonstrating the usage of the reusable integration flows. The example integration flows implement simple process variants. For more information, see *SAP IBP - Reusable Integration Flows*.

You can transfer data between SAP Integrated Business Planning for Supply Chain (SAP IBP) and SAP Cloud Integration using *SAP IBP - Reusable Integration Flows Examples*.

Data integration between SAP IBP and SAP Cloud Integration using the artifacts in the *SAP IBP - Reusable Integration Flows Examples* package has been enabled with SAP IBP 2411 and higher.

Use the *SAP IBP Read - Example* and the *SAP IBP Write - Example* integration flow as a starting point for implementing your custom integration.

3 Prerequisites

You can transfer data if you have configured the connection between SAP Integrated Business Planning for Supply Chain (SAP IBP) and SAP Cloud Integration as follows:

- You've enabled the *Planning - Integration Suite - Cloud Integration Integration* (SAP_COM_0931) communication scenario that allows a connection between SAP IBP and SAP Cloud Integration.
  Basic authentication and authentication with certificate are available for this communication scenario. For more information on the procedure, see *Data Integration Scenarios* > *Data Integration Using SAP Cloud Integration* > *Integration Using SAP IBP - Reusable Integration Flows* > *Setting Up the Integration for Using SAP IBP - Reusable Integration Flows*.
- You've deployed the *SAP IBP - Reusable Integration Flows* package. For more information on the procedure, see *Data Integration Scenarios* > *Data Integration Using SAP Cloud Integration* > *Integration Using SAP IBP - Reusable Integration Flows* > *Setting Up the Integration for Using SAP IBP - Reusable Integration Flows*.
- You've configured SAP IBP data to be available in the system.
  The example is using the SAPIBP1 Planning area with Demand Planning as follows:
  - Time profile
  - Master data: `PRODUCT`, `LOCATION`, `CUSTOMER`
  - Key figure: `FINALDEMANDPLANNINGQTY`
- You've read the about *SAP IBP - Reusable Integration Flows*. For more information, see *Data Integration Scenarios* > *Data Integration Using SAP Cloud Integration* > *Integration Using SAP IBP - Reusable Integration Flows*.

4 SAP IBP Read - Example

4.1 Introduction to SAP IBP Read - Example

This guide showcases the process of reading data from SAP Integrated Planning for Supply Chain (SAP IBP) and demonstrates how to retrieve various data types in packages. The integration flows in the *SAP IBP - Reusable Integration Flows Examples* package can be used to try out the integration between SAP IBP and SAP Cloud Integration.

Examples provided in this guide show how the reusable integration flows can be used and in what order, using basic features of the read process. Running, debugging, and copying the integration flows help you understand the functionality and the proper use of the function. You can try out flow operations, performance, and data accuracy, facilitating error identification and reusability.

Using the *SAP IBP Read - Example* integration flow, you can read the FINALDEMANDPLANNINGQTY (Demand Planning Qty Final) key figure, along with the respective time period, and master data attributes on the WKPRODLOCCUST (technical week, product, location, customer) planning level. For further information on reading data from SAP IBP using reusable integration flows, see *SAP IBP - Reusable Integration Flows*.

Key figure data is retrieved in packages from SAP IBP, and the payload is saved to the attachments of the integration flow run.

4.2 Using SAP IBP Read - Examples

You can set up the *SAP IBP Read - Example* integration flow to read data from SAP IBP, configuring the Content Modifier message bodies and header parameters. The procedure outlines the configuration steps to initialize parameters, fetch data, and close the query when integrating with SAP IBP.

Context

To run the integration flow, perform the following procedure.

Procedure

1. Configure the **Set Initialize Parameter** Content Modifier which sets the input for the SAP IBP Read - Initialize reusable integration flow.

   The message body of the Content Modifier contains the details to call the SAP IBP Read - Initialize reusable integration flow. In the body, you can define, for example, the selection fields or filters for the data you want to read from SAP IBP.

   {·} Sample Code

   ```
   <?xml version="1.0" encoding="UTF-8" standalone="no"?>
   <IBPReads>
       <IBPRead
           Key="${header.IBPReadKey}"
           TypeOfData="KeyFigures"
           PlanningArea="SAPIBP1"
           Destination="${header.IBPReadDestination}"
           TimeAggregationLevel="2"

   Select="PRDID,LOCID,CUSTID,TSTFR,PERIOD_LEVEL_2,FINALDEMANDPLANNINGQTY"
           PackageSizeInMB = "50"
       />
   </IBPReads>
   ```

   Header parameters are used to control the example integration flow behavior and define variables that can be reused throughout the entire integration flow as follows:

Parameter Name	Description
`IBPReadDestination`	Set the `IBPReadDestination` to the SAP IBP system that was previously created in the SAP BTP subaccount.
`IBPReadKey`	In the message header, you can change the value to re-name the payload.
`IBPReadPackageSizeInRows`	Adjust the `IBPReadPackageSizeInRows` value to change the package size in rows.
`IBPReadOffset`	The `IBPReadOffset` value is used for calculating the pay-load when reading the data packages.
`IBPReadFetchFinished`	The `IBPReadFetchFinished` serves as the exit criteria of the Looping Process Call step.
`IBPReadFetchPackageNumber`	The `IBPReadFetchPackageNumber` is used for creating the payload attachments in the Attach Payload step.

   ⓘ Note

   To manage the package size in megabytes, add the `PackageSizeInMB` tag to the message body to calculate the appropriate row number. In the Save Initialize Parameters step, save the value of the

> `IBPRead` response of the `PackageSizeInRows` tag. Overwrite the `PackageSizeInRows` value in the message body of the Set Fetch Parameters step.

2. Configure the header parameters for the **Save Initialize Parameters** Content Modifier.

	Description
`IBPReadRows`	`IBPReadRows` is used later in the XSLT Mapping step to calculate the payload during the data package read.
`IBPReadUUID`	`IBPReadUUID` is responsible for identifying the query when calling *SAP IBP Read - Fetch Data* and *SAP IBP Read - Close* reusable integration flows.
`IBPReadStatus`	`IBPReadStatus` is used in the Router to check if it's responsible for identifying the query when process can continue and start fetching data. It is also used as input for identifying the query when calling the *SAP IBP Read - Close* reusable integration flow.

3. Structure the **Set Fetch Parameters** Content Modifier.

   The **Set Fetch Parameters** sets the input for the *SAP IBP Read - Fetch Data* reusable integration flow using the message body.
4. Create the **Set Close Parameters** body to close the query which sets the input for the *SAP IBP Read - Close* reusable integration flow.
5. Run the integration flow.

   When running an example reusable integration flow, we recommend that you turn on traces to check how the messages are processed. For more information, see Tracing the Execution of an Integration Flow.
6. See the attachments in `MonitorIntegrations` and `APIsMonitor Message Processing`.

   The integration flow also contains an error handling example. See the error handling steps in the *Exception Subprocess*. The *Exception Subprocess* catches exceptions during the run of the integration flow and logs out the content of the last message body in an attachment. If the flow ends with an error status, you can determine the cause of the error in the saved message body.

4.3 Reading Various Data Types

In the message body of the Content Modifier for the Set Initialize Parameters step, you can modify various query initialization parameters as follows.

Reading Key Figure Data

Sample Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<IBPReads>
    <IBPRead Key="Example"
             TypeOfData="KeyFigures"
             PlanningArea="SAPIBP1"
             Destination="IBP_DESTINATION"
             TimeAggregationLevel="2"

Select="PRDID,LOCID,CUSTID,TSTFR,PERIOD_LEVEL_2,FINALDEMANDPLANNINGQTY"/>
</IBPReads>
```

Reading Master Data

Sample Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<IBPReads>
    <IBPRead Key="Example"
             TypeOfData="MASTERDATA"
             MasterDataPrefix="IBP"
             MasterDataCoreType="PRODUCT"
             Destination="IBP_DESTINATION"
             Select="PRDID,PRDDESCR"/>
</IBPReads>
```

Reading Time Profile Data

Sample Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<IBPReads>
    <IBPRead Key="Example"
             TypeOfData="TIMEPROFILE"
             TimeProfile="1"
             Select="PERIODID,DESCR,TSTFR,TSTTO"
             Destination="IBP_DESTINATION"/>
</IBPReads>
```

5 SAP IBP Write - Example

5.1 Introduction to SAP IBP Write - Example

This guide showcases the process of writing data to SAP Integrated Planning for Supply Chain (SAP IBP) and demonstrates how to write various data types in packages within the flow. The *SAP IBP - Reusable Integration Flows Examples* package is used for running, debugging, and copying the integration flows.

The examples provided in this guide show how the reusable integration flows can be used and in what order, using basic features of the write process. Running, debugging, and copying the integration flows help you understand the functionality and proper use of the function. You can try out flow operations, performance, and data accuracy, facilitating error identification and reusability. For further information on writing data to SAP IBP using reusable integration flows, see *SAP IBP - Reusable Integration Flows*.

5.2 Using SAP IBP Write - Examples

You can set up the *SAP IBP Write - Example* integration flows to write data to SAP IBP by configuring the Content Modifier message bodies and header parameters. The procedure outlines the configuration steps to create batches, post data, and process posted data when integrating with SAP IBP.

Context

This example showcases the order and usage of the reusable integration flows, focusing on the write process. You can write `Location Master Data` using the IBPLOCATION master data type, or the ADJDELIVQTY (Delivered Quantity Adjusted) key figure using the LOCPRODCUSTDAILY (location, product, customer, daily) planning level. For further information on writing data to SAP IBP using reusable integration flows, see *SAP IBP - Reusable Integration Flows*.

Master data, and key figure data is written in packages to SAP IBP.

Use the *SAP IBP Write - Example* integration flow as a starting point for implementing your custom integration. To run the integration flow by configuring the Content Modifier, perform the following procedure.

Procedure

1. Configure the **Set Create Batches Parameter** Content Modifier.

   The message body of the Content Modifier contains the details to call the SAP IBP Write - Create Batches reusable integration flow.

   Sample Code

   ```
   1.
   <?xml version="1.0" encoding="UTF-8" standalone="no"?>
   <IBPWriteBatches>
       <IBPWriteBatch
           Key="Example"
           Name="Write Process Example"
           Destination="IBP_DESTINATION"
           Command="INSERT_UPDATE"
           PlanningArea="SAPIBP1"
           PlanningAreaVersion=""
           DetailedTrace=""
           ProcessUnchangedData="FALSE"
       />
   </IBPWriteBatches>
   ```

   Header parameters are used to control the example integration flow behavior and define variables that can be reused throughout the entire integration flow as follows:

Parameter Name	Description
`IsLastPackage`	This flag indicates the last package that will be written, and post processing can be called.
`LoopCounter`	This counter indicates which package will be written.
`DataToWrite`	This header parameter stores the data part which will be written in the current package.
`TypeOfData`	This field defines the type of data to write to SAP IBP. The default upload is key figure. Set the parameter value to `MASTERDATA` if you want to write master data.

2. Call the **Create Batches** step.
3. Configure the **Getting Data** step.

   The Getting Data step is used to create and store the hardcoded data which is going to be written to SAP IBP.

   This step is just an example of how data should be structured and how it can be stored.
4. Call the **Post Data** step.

   The Post Data is called repeatedly in a looping process. The Looping Process Call shows an example logic of how Post Data can be called multiple times for packaging. Configure the message body of the *MD Post Data Parameters* or the *KF Post Data Parameters* Content Modifiers, which contains the details of the batch item and the data to be saved to the SAP IBP staging table.

5. Call the **Process Posted Data** step.
6. Run the integration flow.

   When running an example reusable integration flow, we recommend that you turn on traces to check how the messages are processed. For more information, see Tracing the Execution of an Integration Flow.
7. See the attachments in `MonitorIntegrations` and `APIsMonitor Message Processing`.

   The integration flow also contains an error handling example. See the error handling steps in the *Exception Subprocess*. The *Exception Subprocess* catches exceptions during the run of the integration flow and logs out the content of the last message body in an attachment. If the flow ends with an error status, you can determine the cause of the error in the saved message body.

5.3 Writing Various Data Types

To write master data to IBP, you have to modify the Message Body of the Set Create Batches Parameters and the Set Post Data Parameters Content Modifiers.

Master Data

Set Create Batches Parameters step

Sample Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<IBPWriteBatches>
    <IBPWriteBatch Key="Example"
                   Name="Write Process Example"
                   Destination="IBP_DESTINATION"
                   Command="INSERT_UPDATE"
                   PlanningArea=""
                   PlanningAreaVersion=""
                   DetailedTrace=""
                   ProcessUnchangedData="FALSE"/>
</IBPWriteBatches>
```

Set MD Post Data Parameters step

Sample Code

```
<<IBPWriteMasterData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                     MasterDataPrefix="IBP"
                     MasterDataType="LOCATION"
                     FileName="Location File"
                     FieldList="LOCID,LOCTYPE,LOCDESCR"
                     BatchKey="Example">
    <item>
        <LOCID>1010</LOCID>
        <LOCTYPE>P</LOCTYPE>
        <LOCDESCR>Plant 1 DE</LOCDESCR>
    </item>
```

```
</IBPWriteMasterData>
```

Key Figure

Set Create Batches Parameters step

Sample Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<IBPWriteBatches>
    <IBPWriteBatch Key="Example"
                   Name="Write Process Example"
                   Destination="IBP_DESTINATION"
                   Command="INSERT_UPDATE"
                   PlanningArea="SAPIBP1"
                   PlanningAreaVersion=""
                   DetailedTrace=""
                   ProcessUnchangedData="FALSE"/>
</IBPWriteBatches>
```

Set KF Post Data Parameters step

Sample Code

```
<IBPWriteKeyFigures xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    FieldList="PRDID,LOCID,CUSTID,KEYFIGUREDATE,ADJDELIVQTY"
                    BatchKey="Example"
                    FileName="Key Figure File"
                    TimeDisaggregationLevel="1">
    <item>
        <PRDID>IBP-100</PRDID>
        <LOCID>1010</LOCID>
        <CUSTID>DUMMY</CUSTID>
        <KEYFIGUREDATE>2024-11-01</KEYFIGUREDATE>
        <ADJDELIVQTY>1</ADJDELIVQTY>
    </item>
</IBPWriteKeyFigures>
```

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