Data Migration: Inventory Balances
SAP Business One to SAP S/4HANA Cloud Public Edition Guide

For SAP Cloud Integration

Version 1.0.1 | December 2025

Data Transformation: Inventory Balances SAP Business One to SAP S/4HANA Cloud Public Edition

1. Introduction

In this guide you shall learn how to consume the integration flow to transform data of an object in SAP Business one into a form suitable and acceptable by SAP S/4HANA Cloud Public Edition.

1.1. Definition

This integration flow is an accelerator that transforms the standard data of 'Inventory Balances' object in SAP Business One into the SAP S/4HANA Cloud Public Edition's migration template of object 'Inventory Balances.

1.2. Intended Audience

This integration flow is intended to be used by both partners and customers who are in the data migration phase of their implementation project of moving their SAP Business One system to SAP S/4HANA Cloud Public Edition system. This integration flow shall act as an accelerator to kick start and speed up your data migration task. However, it does not cover all attributes and mappings, hence enhance it as per your requirement.

1.3. Structure

The structure of this guide follows the sequence of steps required to consume the integration flow for the purpose of data transformation using SAP Integration Suite service on SAP BTP.

1.4. System Connectivity

To use this integration flow, you don't have to connect your SAP Business One system or SAP S/4HANA Cloud system to SAP Integration Suite. The data extracted from SAP Business One is passed to SAP Integration Suite over HTTPS and the data prepared as output is a file that can be taken manually and uploaded in SAP S/4HANA Cloud Public Edition via Migration Cockpit.

1.5. Additional Documentation

You may refer to these topics via links to gain insights into various topics you need to know before using the integration flow.

SAP Integration Suite
https://help.sap.com/docs/cloud-integration/sap-cloud-integration/sap-cloud-integration?version=Cloud

SAP S/4HANA Cloud Public Edition Migration Cockpit
https://help.sap.com/docs/SAP_S4HANA_CLOUD/d5699934e7004d048c4801b552f3b013/f32db0c240484241abc53a876253e118.html?version=2408.500

Technical Data Migration to SAP S/4HANA Cloud
https://learning.sap.com/videos/overseeing-the-transition-from-sap-business-bydesign-to-sap-s-4hana-cloud-

Setting up Required systems and Users
https://learning.sap.com/videos/preparing-for-data-migration-from-sap-business-bydesign-to-sap-s-4hana-cloud

2. Business Scenario

Customers who are looking to migrate from SAP Business One ERP system to SAP S/4HANA Cloud Public Edition ERP system need an easy data migration path to save time and effort on data migration activities which is quite intensive.

With this intention SAP has delivered the mapping of data fields from SAP Business One Inventory object to SAP S/4HANA Cloud Public Edition equivalent object as part of this integration flow. The end result of the integration flow will provide you the Inventory Balances file that is in the form of SAP S/4HANA Cloud Public Edition's product migration template. You shall upload this file via Migration Cockpit in SAP S/4HANA Cloud Public Edition. This helps you to save effort and prepare the file easily for upload.

3. Prerequisites

To consume the integration flow, you should have the following things done
   a) Set up account in SAP BTP and subscribe to SAP Integration Suite service.
   b) Set up required user roles and authorization in SAP BTP subaccount
   c) Set up the required capabilities in SAP Integration Suite to trigger integration flow.
   d) User should have authorisation to the Migration Cockpit app in SAP S/4HANA Cloud Public Edition
   e) The user should have access to the Inventory module in SAP Business One system.
   f) User should use the date format **DD.MM.YYYY** when extracting from SAP Business One system
   g) User should use the Query Generator in SAP Business One to extract the data.
   h) The data obtained from Query Generator should be exported in the format *.xlsx

Please refer to the links mentioned in the section Additional Documentation to complete the prerequisites.

4. Consumption of integration flow

Follow the steps below to transform your data and upload to SAP S/4HANA Cloud Public Edition.

4.1. Data Extraction

To consume the integration flow for transformation you need to extract the required Inventory data from SAP Business One system.

Extract the data from SAP Business One: Inventory Balances data is extracted from the SAP Business One using the following approach.

1. Run the below query in SAP Business One system. This query collects data from all relevant tables where the inventory data is stored.

   HANA DB:

   SELECT A.* FROM (SELECT T1."ItemCode" AS "Material", CAST(T1."Location" AS VARCHAR(50)) AS "Plant", T1."Warehouse" AS "StorageLocation", SUM(T1."InQty"-T1."OutQty") AS "EntryQuantity", MAX(T0."InvntryUom") AS "EntryUnitofMeasure", SUM(T1."TransValue") AS "ExternalAmount(LC)",'' AS "BatchNumber", '' AS "ExpirationDate", '' AS "ManufacturingDate", '' AS "SerialNumber" FROM OITM T0 INNER JOIN OINM T1 ON T0."ItemCode"=T1."ItemCode" WHERE T0."ManBtchNum"='N' AND T0."ManSerNum"='N' AND T0."ItemType"='I' GROUP BY T1."ItemCode", T1."Warehouse", T1."Location") A WHERE A."EntryQuantity"<>0.00 UNION ALL SELECT B."Material",B."Plant",B."StorageLocation",B."EntryQuantity",B."EntryUnitofMeasure", (B."StockValue"/B."InStock")*B."EntryQuantity" AS "ExternalAmount(LC)", B."BatchNumber", B."ExpirationDate", B."ManufacturingDate", B."SerialNumber" FROM (SELECT T1."ItemCode" AS "Material", CAST(T1."Location" AS VARCHAR(50)) AS "Plant", T1."Warehouse" AS "StorageLocation", MAX(T2."Quantity") AS "EntryQuantity", SUM(T1."InQty"-T1."OutQty") AS "InStock", SUM(T1."TransValue") AS "StockValue",MAX(T0."InvntryUom") AS "EntryUnitofMeasure", T2."BatchNum" AS "BatchNumber",FORMAT(CAST(T2."ExpDate" AS date) ,'dd.MM.yyyy') AS "ExpirationDate", FORMAT(CAST(T2."PrdDate" AS date) ,'dd.MM.yyyy') AS "ManufacturingDate", '' AS "SerialNumber" FROM OITM T0 INNER JOIN OINM T1 ON T0."ItemCode"=T1."ItemCode" INNER JOIN OIBT T2 ON T1."ItemCode"=T2."ItemCode" AND T1."Warehouse"=T2."WhsCode" WHERE T0."ManBtchNum"='Y' AND T0."ManSerNum"='N' AND T0."ItemType"='I' AND T2."Quantity"<>0.00 GROUP BY T1."ItemCode", T1."Warehouse", T1."Location", T2."BatchNum",T2."ExpDate",T2."PrdDate") B WHERE B."InStock"<>0.00 UNION ALL SELECT B."Material",B."Plant",B."StorageLocation",B."EntryQuantity",B."EntryUnitofMeasure", (B."StockValue"/B."InStock")*B."EntryQuantity" AS "ExternalAmount(LC)", B."BatchNumber", B."ExpirationDate", B."ManufacturingDate", B."SerialNumber" FROM (SELECT T1."ItemCode" AS "Material", CAST(T1."Location" AS VARCHAR(50)) AS "Plant", T1."Warehouse" AS "StorageLocation", MAX(T3."Quantity") AS "EntryQuantity", SUM(T1."InQty"-T1."OutQty") AS "InStock", SUM(T1."TransValue") AS "StockValue",MAX(T0."InvntryUom") AS "EntryUnitofMeasure", '' AS "BatchNumber", '' AS "ExpirationDate", '' AS "ManufacturingDate", T3."DistNumber" AS "SerialNumber" FROM OITM T0 INNER JOIN OINM T1 ON T0."ItemCode"=T1."ItemCode" INNER JOIN (SELECT T4."ItemCode", T4."WhsCode", T5."DistNumber", T4."Quantity" FROM OSRQ T4 INNER JOIN OSRN T5 ON T4."MdAbsEntry"=T5."AbsEntry" WHERE T4."Quantity"<>0.00) T3 ON T1."ItemCode"=T3."ItemCode" AND T1."Warehouse"=T3."WhsCode" WHERE T0."ManBtchNum"='N' AND T0."ManSerNum"='Y' AND T0."ItemType"='I' AND T3."Quantity"<>0.00 GROUP BY T1."ItemCode", T1."Warehouse", T1."Location",T3."DistNumber") B WHERE B."InStock"<>0.00

   SQL DB:

   SELECT A.* FROM (SELECT T1."ItemCode" AS "Material", CAST(T1."Location" AS VARCHAR(50)) AS "Plant", T1."Warehouse" AS "StorageLocation", SUM(T1."InQty"-T1."OutQty") AS "EntryQuantity", MAX(T0."InvntryUom") AS "EntryUnitofMeasure", SUM(T1."TransValue") AS "ExternalAmount(LC)",'' AS "BatchNumber", '' AS "ExpirationDate", '' AS "ManufacturingDate", '' AS "SerialNumber" FROM OITM T0 INNER JOIN OINM T1 ON T0."ItemCode"=T1."ItemCode" WHERE T0."ManBtchNum"='N' AND T0."ManSerNum"='N' AND T0."ItemType"='I' GROUP BY T1."ItemCode", T1."Warehouse",

T1."Location") A WHERE A."EntryQuantity"<>0.00 UNION ALL SELECT B."Material",B."Plant",B."StorageLocation",B."EntryQuantity",B."EntryUnitofMeasure", (B."StockValue"/B."InStock")*B."EntryQuantity" AS "ExternalAmount(LC)", B."BatchNumber", B."ExpirationDate", B."ManufacturingDate", B."SerialNumber" FROM (SELECT T1."ItemCode" AS "Material", CAST(T1."Location" AS VARCHAR(50)) AS "Plant", T1."Warehouse" AS "StorageLocation", MAX(T2."Quantity") AS "EntryQuantity", SUM(T1."InQty"-T1."OutQty") AS "InStock", SUM(T1."TransValue") AS "StockValue",MAX(T0."InvntryUom") AS "EntryUnitofMeasure", T2."BatchNum" AS "BatchNumber",FORMAT(CAST(T2."ExpDate" AS date) ,'dd.MM.yyyy') AS "ExpirationDate", FORMAT(CAST(T2."PrdDate" AS date) ,'dd.MM.yyyy') AS "ManufacturingDate", '' AS "SerialNumber" FROM OITM T0 INNER JOIN OINM T1 ON T0."ItemCode"=T1."ItemCode" INNER JOIN OIBT T2 ON T1."ItemCode"=T2."ItemCode" AND T1."Warehouse"=T2."WhsCode" WHERE T0."ManBtchNum"='Y' AND T0."ManSerNum"='N' AND T0."ItemType"='I' AND T2."Quantity"<>0.00 GROUP BY T1."ItemCode", T1."Warehouse", T1."Location", T2."BatchNum",T2."ExpDate",T2."PrdDate") B WHERE B."InStock"<>0.00 UNION ALL SELECT B."Material",B."Plant",B."StorageLocation",B."EntryQuantity",B."EntryUnitofMeasure", (B."StockValue"/B."InStock")*B."EntryQuantity" AS "ExternalAmount(LC)", B."BatchNumber", B."ExpirationDate", B."ManufacturingDate", B."SerialNumber" FROM (SELECT T1."ItemCode" AS "Material", CAST(T1."Location" AS VARCHAR(50)) AS "Plant", T1."Warehouse" AS "StorageLocation", MAX(T3."Quantity") AS "EntryQuantity", SUM(T1."InQty"-T1."OutQty") AS "InStock", SUM(T1."TransValue") AS "StockValue",MAX(T0."InvntryUom") AS "EntryUnitofMeasure", '' AS "BatchNumber", '' AS "ExpirationDate", '' AS "ManufacturingDate", T3."DistNumber" AS "SerialNumber" FROM OITM T0 INNER JOIN OINM T1 ON T0."ItemCode"=T1."ItemCode" INNER JOIN (SELECT T4."ItemCode", T4."WhsCode", T5."DistNumber", T4."Quantity" FROM OSRQ T4 INNER JOIN OSRN T5 ON T4."MdAbsEntry"=T5."AbsEntry" WHERE T4."Quantity"<>0.00) T3 ON T1."ItemCode"=T3."ItemCode" AND T1."Warehouse"=T3."WhsCode" WHERE T0."ManBtchNum"='N' AND T0."ManSerNum"='Y' AND T0."ItemType"='I' AND T3."Quantity"<>0.00 GROUP BY T1."ItemCode", T1."Warehouse", T1."Location",T3."DistNumber") B WHERE B."InStock"<>0.00

2. Export the result of the query in an excel file and name it "Inventory.xlsx" . Make sure that you do not use any other name for the file.

4.2. Data Preparation

The data extracted is in raw form and cannot be consumed by the integration flow. Therefore, you should prepare the data in the form that integration flow can accept.

Prepare the extracted data: Formatting and merging is required to be done as part of preparation of the data for iflow.

1. The output excel file should be exactly as shown below. DO NOT CHANGE the column names (highlighted in yellow)

2. Zip the excel file to create a *.ZIP file. Save this file.

4.3. Data Transformation

Once the data is prepared and ready in the required form, you are ready to execute the integration flow.

Make the following preparations to load the integration flow for executing data transformation.

I. Download the iflow for Inventory Balances B1 (provided by SAP).
II. Upload the iflow in your SAP Integration Suite system.
III. Deploy the integration flow.
IV. After successful deployment of integration flow, make a note of the End Point URL generated.

To consume the integration flow for data transformation and create a S/4HANA migration file, follow the steps below. For details refer to the [learning session](#).

I. Set up an application to trigger the integration flow e.g., Postman.
II. Create a POST HTTP request, provide the user credentials, and upload the ZIP file (created in Data Preparation step) in the body of the request.
III. Send the request to the end point URL of integration flow. This will trigger the integration flow and start the data transformation. As a result, you shall receive a response file back from your triggering application.
IV. Save the response ZIP file received. Extract the ZIP package and Open the XML file in Microsoft Excel. Click on *Save*.

Note: In the response XML, some records contain fields that require your attention. These instances can be identified by locating cells with the value **"<<Enter Manually>>"**.

4.4. Data Upload

Once the data is transformed and you have received the S/4HANA template file filled with the data (response file) you are ready to upload the same to S/4HANA system via Migration Cockpit.

Follow the below steps to upload the data to S/4HANA system:

a. Save the response ZIP file received.

b. Extract the ZIP package and open the XML file in Microsoft Excel. Click on save. Close the file.

c. This is the file that should be taken to SAP S/4HANA Cloud Public Edition migration project and uploaded. For details on uploading a file via migration cockpit in SAP S/4HANA Cloud Public Edition refer to this [help document](#).

5. Customization and Enhancement of integration flow

You can customize the mapping or enhance the integration flow as per your requirement. For details on customization, you can refer to these tutorials. These tutorials are built for SAP Business ByDesign but the same concept applies to SAP Business One object iflows.

- [Tutorial 1](#): Customizing standard integration flows for data migration from SAP Business ByDesign to SAP S/4HANA Cloud Public Edition.
- [Tutorial 2](#): Updating data structure in integration flows for data migration from SAP Business ByDesign to SAP S/4HANA Cloud Public Edition.

6. Contact Information

Email:
DL_66604530AF3F4B0136989F8F@global.corp.sap

Support Ticket Component:
CA-S4H-B1-IFLOW