Data Migration: Open Sales Orders
SAP Business One to SAP S/4HANA Cloud Public Edition Guide

For SAP Cloud Integration

Version 1.0.1 | December 2025

Data Transformation: Open Sales Orders 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 'Sales Order' object in SAP Business One into the SAP S/4HANA Cloud Public Edition's migration template of object 'Sales Order'.

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 Sales Order 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 Sales Order 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 Sales orders 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 Sales Orders data from SAP Business One system.

Extract the data from SAP Business One: Open Sales Order data is extracted from the SAP Business One using the following approach.

1. Run the below queries one by one in SAP Business One system. These queries collect data from all relevant tables where Sales order data is stored.
    a) Sales order Header Data

    SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", T1."NumAtCard" AS "CustomerPONum", T1."DocDueDate" AS "RequestedDeliveryDate", T1."DocDate" AS "PricingDate", T1."DocRate" AS "ExchangeRate", T1."TaxDate" AS "DocumentDate", (SELECT T2."PymntGroup" FROM OCTG T2 WHERE T2."GroupNum"=T1."GroupNum") AS "PaymentTerms", T1."Project" AS "WBSElement"
    FROM ORDR T1 WHERE T1."DocStatus"='O'

    b) Sales order Partner Data

    HANA Database Query:

    *SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", ((T1."LineNum"+1)*10) AS "SalesDocItem",*
    T2."CardCode" AS "SoldToParty", T2."CardCode" AS "Ship To Party", T2."ShipToCode" AS "Ship-To Address Code", T2."Address2" AS "Ship-To Address",
    T3."StreetS" AS "Ship-To Street", T3."BlockS" AS "Ship-To Block", T3."BuildingS" AS "Ship-To Building", T3."CityS" AS "Ship-To City", T3."ZipCodeS" AS "Ship-To ZipCode", (SELECT ST."Name" FROM OCST ST WHERE ST."Country"=T3."CountryS" AND ST."Code"=T3."StateS") AS "Ship-To State", (SELECT CR."Name" FROM OCRY CR WHERE CR."Code"=T3."CountryS") AS "Ship-To Country", T3."CountyS" AS "Ship-To Region", CASE WHEN T2."FatherType"='D' THEN T2."FatherCard" ELSE T2."CardCode" END AS "Bill To Party", CASE WHEN T2."FatherType"='P' AND T2."FatherCard" IS NOT NULL THEN T2."FatherCard" ELSE T2."CardCode" END AS "Payer",
    (SELECT SE."SlpName" FROM OSLP SE WHERE SE."SlpCode"=T2."SlpCode") AS "Sales Employee", (SELECT HR."firstName"||' '||HR."lastName" FROM OHEM HR WHERE HR."empID"=T2."OwnerCode") AS "Employee Responsible", (SELECT CP."Name" FROM OCPR CP WHERE CP."CntctCode"=T2."CntctCode") AS "Contact Person ID", (SELECT UR."U_NAME" FROM OUSR UR WHERE UR."USERID"=T2."UserSign") AS "User Responsibile", T2."CardCode" AS "CustomerNumber", T2."PayToCode" AS "Bill-To Address Code", T2."Address" AS "Bill-To Address", T3."StreetB" AS "Bill-To Street", T3."BlockB" AS "Bill-To Block", T3."BuildingB" AS "Bill-To Building", T3."CityB" AS "Bill-To City", T3."ZipCodeB" AS "Bill-To ZipCode", (SELECT ST."Name" FROM OCST ST WHERE ST."Country"=T3."CountryB" AND ST."Code"=T3."StateB") AS "Bill-To State", (SELECT CR."Name" FROM OCRY CR WHERE CR."Code"=T3."CountryB") AS "Bill-To Country", T3."CountyB" AS "Bill-To Region", T4."Phone1" AS "TelephoneNum", T4."Fax", T3."BpGSTN" AS "Cust_VATRegNum"
    FROM RDR1 T1 INNER JOIN ORDR T2 ON T1."DocEntry"=T2."DocEntry"
    INNER JOIN RDR12 T3 ON T2."DocEntry"=T3."DocEntry" AND T2."ObjType"=T3."ObjectType"
    INNER JOIN OCRD T4 ON T2."CardCode"=T4."CardCode"
    WHERE T2."DocStatus"='O' AND T1."LineStatus"='O'

    SQL Database Query:

    *SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", ((T1."LineNum"+1)*10) AS "SalesDocItem",*
    T2."CardCode" AS "SoldToParty", T2."CardCode" AS "Ship To Party", T2."ShipToCode" AS "Ship-To Address Code", T2."Address2" AS "Ship-To Address",
    T3."StreetS" AS "Ship-To Street", T3."BlockS" AS "Ship-To Block", T3."BuildingS" AS "Ship-To Building", T3."CityS" AS "Ship-To City", T3."ZipCodeS" AS "Ship-To ZipCode", (SELECT ST."Name" FROM OCST ST WHERE ST."Country"=T3."CountryS" AND ST."Code"=T3."StateS") AS "Ship-To State", (SELECT CR."Name" FROM OCRY CR WHERE CR."Code"=T3."CountryS") AS "Ship-To Country", T3."CountyS"

```
AS "Ship-To Region", CASE WHEN T2."FatherType"='D' THEN T2."FatherCard" ELSE T2."CardCode"
END AS "Bill To Party", CASE WHEN T2."FatherType"='P' AND T2."FatherCard" IS NOT NULL THEN
T2."FatherCard" ELSE T2."CardCode" END AS "Payer",
(SELECT SE."SlpName" FROM OSLP SE WHERE SE."SlpCode"=T2."SlpCode") AS "Sales Employee",
(SELECT HR."firstName" + ' ' + HR."lastName" FROM OHEM HR WHERE
HR."empID"=T2."OwnerCode") AS "Employee Responsible", (SELECT CP."Name" FROM OCPR CP
WHERE CP."CntctCode"=T2."CntctCode") AS "Contact Person ID", (SELECT UR."U_NAME" FROM
OUSR UR WHERE UR."USERID"=T2."UserSign") AS "User Responsibile", T2."CardCode" AS
"CustomerNumber", T2."PayToCode" AS "Bill-To Address Code", T2."Address" AS "Bill-To Address",
T3."StreetB" AS "Bill-To Street", T3."BlockB" AS "Bill-To Block", T3."BuildingB" AS "Bill-To Building",
T3."CityB" AS "Bill-To City", T3."ZipCodeB" AS "Bill-To ZipCode", (SELECT ST."Name" FROM OCST ST
WHERE ST."Country"=T3."CountryB" AND ST."Code"=T3."StateB") AS "Bill-To State", (SELECT
CR."Name" FROM OCRY CR WHERE CR."Code"=T3."CountryB") AS "Bill-To Country", T3."CountyB"
AS "Bill-To Region", T4."Phone1" AS "TelephoneNum", T4."Fax", T3."BpGSTN" AS "Cust_VATRegNum"
FROM RDR1 T1 INNER JOIN ORDR T2 ON T1."DocEntry"=T2."DocEntry"
INNER JOIN RDR12 T3 ON T2."DocEntry"=T3."DocEntry" AND T2."ObjType"=T3."ObjectType"
INNER JOIN OCRD T4 ON T2."CardCode"=T4."CardCode"
WHERE T2."DocStatus"='O' AND T1."LineStatus"='O'
```

c) Sales order Item Data

```
SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", ((T1."LineNum"+1)*10) AS
"SalesDocItem", T1."ItemCode" AS "ProductNumber", T1."unitMsr" AS "SalesUnit",
CAST(T1."LocCode" AS VARCHAR(50))  AS "Plant", T1."WhsCode" AS "StorageLocation",
T1."Dscription" AS "ItemDescription", T1."Project" AS "WBSElement" FROM RDR1 T1 INNER JOIN
ORDR T2 ON T1."DocEntry"=T2."DocEntry" WHERE T2."DocStatus"='O' AND T1."LineStatus"='O'
```

d) Sales order Header Texts

HANA Database Query:

```
SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", (SELECT T2."ShortName"
FROM OLNG T2 WHERE T2."Code"=T1."LangCode") AS "LanguageKey", T1."Comments" AS
"GeneralRemarks", T1."Header" AS "OpeningRemarks", T1."Footer" AS "ClosingRemarks" FROM
ORDR T1 WHERE T1."DocStatus"='O' AND IFNULL(CAST(T1."Comments" AS VARCHAR),'')<>''
```

SQL Database Query:

```
SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", (SELECT T2."ShortName"
FROM OLNG T2 WHERE
T2."Code"=T1."LangCode") AS "LanguageKey", T1."Comments" AS "GeneralRemarks", T1."Header"
AS "OpeningRemarks", T1."Footer" AS "ClosingRemarks" FROM ORDR T1 WHERE
T1."DocStatus"='O' AND ISNULL(CAST(T1."Comments" AS VARCHAR),'')<>''
```

e) Sales order Item Texts

HANA Database Query:

```
SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", ((T1."LineNum"+1)*10) AS
"SalesDocItem", (SELECT T3."ShortName" FROM OLNG T3 WHERE T3."Code"=T2."LangCode") AS
"LanguageKey", T1."Text" FROM RDR1 T1 INNER JOIN ORDR T2 ON T1."DocEntry"=T2."DocEntry"
WHERE T2."DocStatus"='O' AND T1."LineStatus"='O'
AND IFNULL(CAST(T1."Text" AS VARCHAR),'')<>''
```

SQL Database Query:

```
SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", ((T1."LineNum"+1)*10) AS
"SalesDocItem",
```

(SELECT T3."ShortName" FROM OLNG T3 WHERE T3."Code"=T2."LangCode") AS "LanguageKey", T1."FreeTxt" AS "Text"
FROM RDR1 T1 INNER JOIN ORDR T2 ON T1."DocEntry"=T2."DocEntry" WHERE T2."DocStatus"='O' AND T1."LineStatus"='O'
AND CAST(T1."FreeTxt" AS VARCHAR) is not NULL and CAST(T1."FreeTxt" AS VARCHAR) <>''

f) Sales order Item Conditions

*SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc",((T1."LineNum"+1)*10) AS "SalesDocItem", T1."PriceBefDi",T1."DiscPrcnt", T1."PriceBefDi"-T1."Price" AS "Discount Amount per Row", T1."Price", T1."Quantity",T1."LineTotal",T1."VatPrcnt", T1."PriceAfVAT",T1."VatSum", T1."GTotal", F1."Freight1", F1."Freight1Tax%", F1."Freight1TaxAmt", F2."Freight2", F2."Freight2Tax%", F2."Freight2TaxAmt", F3."Freight3", F3."Freight3Tax%", F3."Freight3TaxAmt", T1."Currency" FROM RDR1 T1 INNER JOIN ORDR T2 ON T1."DocEntry"=T2."DocEntry"*
LEFT JOIN (SELECT T1."DocEntry", T1."LineNum", T1."LineTotal" AS "Freight1", T1."VatPrcnt" AS "Freight1Tax%", T1."VatSum" AS "Freight1TaxAmt" FROM RDR2 T1 WHERE T1."GroupNum"='0') F1 ON T1."DocEntry"=F1."DocEntry" AND T1."LineNum"=F1."LineNum"
LEFT JOIN (SELECT T1."DocEntry", T1."LineNum", T1."LineTotal" AS "Freight2", T1."VatPrcnt" AS "Freight2Tax%", T1."VatSum" AS "Freight2TaxAmt" FROM RDR2 T1 WHERE T1."GroupNum"='1') F2 ON T1."DocEntry"=F2."DocEntry" AND T1."LineNum"=F2."LineNum"
LEFT JOIN (SELECT T1."DocEntry", T1."LineNum", T1."LineTotal" AS "Freight3", T1."VatPrcnt" AS "Freight3Tax%", T1."VatSum" AS "Freight3TaxAmt" FROM RDR2 T1 WHERE T1."GroupNum"='2') F3 ON T1."DocEntry"=F3."DocEntry" AND T1."LineNum"=F3."LineNum"
WHERE T2."DocStatus"='O' AND T1."LineStatus"='O' ORDER BY T1."DocEntry", T1."LineNum"

g) Sales order Header Conditions

SELECT S0."LegacySalesDoc",S1."NetAmount",S0."DiscPrcnt", S0."DiscSum",S0."TotalExpns",S0."VatSum",S0."DocTotal", S0."DocCur"
FROM (SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", T1."DocStatus", T1."DiscPrcnt", T1."DiscSum", T1."TotalExpns", T1."VatSum", T1."DocTotal", T1."DocCur" FROM ORDR T1) S0
INNER JOIN (SELECT T1."DocEntry", SUM(T2."LineTotal") AS "NetAmount" FROM ORDR T1 INNER JOIN RDR1 T2 ON T1."DocEntry"=T2."DocEntry" GROUP BY T1."DocEntry") S1 ON S0."LegacySalesDoc"=S1."DocEntry" WHERE S0."DocStatus"='O'

h) Sales order Schedule Lines

HANA Database Query:

*SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", ((T1."LineNum"+1)*10) AS "SalesDocItem", T3."SchdLine" AS "DeliverySchLineNum", T1."Quantity" AS "OrderQty", T1."OpenQty", T1."unitMsr",T3."CfmQty" AS "ScheduledQty", (SELECT IT."InvntryUom" FROM OITM IT WHERE IT."ItemCode"=T1."ItemCode") AS "ScheduledQty Unit" , IfNull(T3."CfmDate",T2."DocDueDate") AS "DeliveryDate" FROM RDR1 T1 INNER JOIN ORDR T2 ON T1."DocEntry"=T2."DocEntry" LEFT JOIN OSLD T3 ON T1."DocEntry"=T3."DocEntry" AND T1."LineNum"=T3."DocLineNum" AND T1."ObjType"=T3."ObjType" WHERE T2."DocStatus"='O' AND T1."LineStatus"='O' ORDER BY T2."DocEntry"*

SQL Database Query:

*SELECT CAST(T1."DocEntry" AS VARCHAR(50)) AS "LegacySalesDoc", ((T1."LineNum"+1)*10) AS "SalesDocItem",*
T3."SchdLine" AS "DeliverySchLineNum", T1."Quantity" AS "OrderQty", T1."OpenQty", T1."unitMsr",T3."CfmQty" AS "ScheduledQty", (SELECT IT."InvntryUom" FROM OITM IT WHERE IT."ItemCode"=T1."ItemCode") AS "ScheduledQty Unit" , ISNull(T3."CfmDate",T2."DocDueDate") AS "DeliveryDate" FROM RDR1 T1 INNER JOIN ORDR T2 ON T1."DocEntry"=T2."DocEntry" LEFT JOIN OSLD T3 ON T1."DocEntry"=T3."DocEntry" AND T1."LineNum"=T3."DocLineNum" AND T1."ObjType"=T3."ObjType" WHERE T2."DocStatus"='O' AND T1."LineStatus"='O' ORDER BY T2."DocEntry"

2. Export the result of each query in an excel file, thereby generating multiple excel files, that contains data of sales orders in different views. Name each of the file as mentioned below. **Make sure that you do not use any other name**.
    a. Sales order Header Data: *Header.xlsx*
    b. Sales order Partner Data: *Partner.xlsx*
    c. Sales order Item Data: *Item.xlsx*
    d. Sales order Header Texts: *HeaderTexts.xlsx*
    e. Sales order Item texts: *ItemTexts.xlsx*
    f. Sales order Item Conditions: *ItemConditions.xlsx*
    g. Sales order Header Conditions: *HeaderConditions.xlsx*
    h. Sales order Schedule lines: *ScheduleLineData.xlsx*

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 of various files is required to be done as part of preparation of the data for iflow.

1. Each output excel file should be in the form as shown below. DO NOT CHANGE the column names (highlighted in yellow). For example the Sales Order Header file (Header.xlsx) should look like this:

	A	B	C	D	E	F	G	H	I
1	#	LegacySalesDoc	CustomerPONun	RequestedDeliveryDate	PricingDat	Exchange	DocumentDate	PaymentTerms	WBSElement
2	1	4403		30/06/2024	14/06/202	1	14/06/2024	30 Days	
3	2	4398		30/06/2024	13/06/202	1	13/06/2024	30 Days	
4	3	4399		30/06/2024	13/06/202	1	13/06/2024	30 Days	
5	4	4406		30/06/2024	14/06/202	1	14/06/2024	30 Days	
6	5	4445		31/08/2024	09/08/202	1	09/08/2024	30 Days	
7	6	4486	4510118445	31/10/2024	11/10/202	1	11/10/2024	30 Days	
8	7	4496		30/11/2024	03/11/202	1	30/11/2024	2 Days	
9	8	4482		31/10/2024	07/10/202	1	07/10/2024	30 Days	
10	9	4501		08/11/2024	06/11/202	1	06/11/2024	30 Days	
11	10	4502		06/11/2024	06/11/202	1	06/11/2024	30 Days	
12	11	4503		07/11/2024	06/11/202	1	06/11/2024	30 Days	
13	12	4504		07/11/2024	06/11/202	1	06/11/2024	30 Days	
14	13	4521		22/11/2024	22/11/202	1	22/11/2024	30 Days	
15	14	4525		06/12/2024	06/12/202	1	06/12/2024	30 Days	
16	15	4526		06/12/2024	06/12/202	1	06/12/2024	30 Days	
17	16	4528		19/12/2024	19/12/202	1	19/12/2024	30 Days	
18	17	4533		19/01/2025	09/01/202	1	09/01/2025	30 Days	
19	18	4532		17/01/2025	10/01/202	1	10/01/2025	30 Days	
20	19	4534		20/01/2025	10/01/202	1	10/01/2025	30 Days	
21	20	4535		25/01/2025	10/01/202	1	10/01/2025	30 Days	

2. Zip all the excel files to create one *.ZIP file and save it. This zip file is the payload for the iflow.

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 Sales Order for 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 might 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