Building block Configuration Guide

CUSTOMER

Commercial Planning Send IBP for Demand baseline to SAP Analytics Cloud Sales model
May 2023
English

Commercial Planning Send SAP IBP for Demand baseline to Analytics Cloud Sales model

Contents

1 Prerequisites

The package Commercial Planning contains SAP Analytic Cloud models for Sales and Marketing Planning, as well as corresponding SAP Integration Suite Integration Flows. These Integration Flows read (baseline quantities) data from IBP, read prices from SAP S/4HANA to write them into SAP Analytic Cloud. There are also Integration Flows to write the (planned drivers) data from SAP Analytic Cloud to SAP IBP for demand.

The Integration Flow “Send IBP for Demand amounts to Analytics Cloud Sales model” connects the content package model for Sales Planning in SAP Analytic Cloud with SAP IBP for Demand. This flow sends the baseline from IBP into SAC.

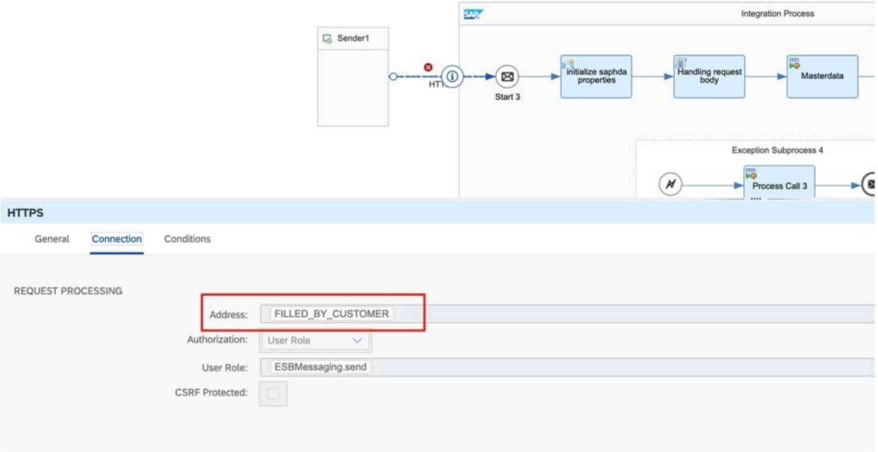
This Integration Flow is a possible implementation approach. But it is necessary to check the individual business needs.

2 Documentation

The Integration Flow reads fact data from IBP for demand, transforms the data and writes the data into SAP Analytics Cloud, into the sales planning model of the content package Commercial Planning. To minimize the memory footprint, a semantical partitioning on the timestamp is used. So data is read month by month, transformed month by month and written into SAP Analytic cloud month by month. As the data is aggregated, it is relevant to have all AMOUNT values for each property combination send to IBP in one loop.

2.1 Starting the flow

The Integration Flow is stated via API call. Externalized Parameter <SAPHDA_API_ENDPOINT>. The Externalized Parameter could be defined as /write2sac so that the Integration Flow can be called via the URL that can be found in the CI Monitor section.



Address: FILLED_BY_CUSTOMER

Authorization: User Role

User Role: ESBMessaging.send

CSRF Protected: □

Figure 1 API endpoint definition

The payload that is expected to be sent with this call contains the SAP Analytics Cloud model ID, a date range from when data is read as well as the time horizon that should be exported (calmonthFrom and calmonthTo are included int the boundaries).

An example payload looks like:

```
{
  "modelID": "Cepk9k03peaigaeka98s7gqb32q",
  "calmonthFrom" : "202301",
  "calmonthTo" : "202401"
}
```

2.2 Reading Master Data

The Integration Flow utilizes the following master data

- Customer <-> Salesorganisation Mapping. This mapping has to be stored in the groovy script method CustomerSalesOrg of script saphda_logic.groovy.
- Salesorganisation <-> Company Code. This mapping is read from the S/4HANA API api_salesorganization_srv/A_SalesOrganization
- The timestamps for the semantical partitioning are read from IBP API.

2.3 Reading data from SAP IBP for Demand

Fact data is read from SAP Analytic Cloud via oData API with the query filter stored in property saphda_queryParameters_calculated.

By delivery the select is configured to

- PRDID Product
- CUSTID Customer
- LOCID Location
- BASEFCSTQTY Quantity
- PERIODID3_TSTAMP Timestamp/Month

The filter is set to

- UOMTOID = EA
- PERIODID3_TSTAMP = is filled automatically in the function calcCurrentMonth of groovy script saphda_logic.groovy

2.4 Transformation

The transformation is processed in the method transform of groovy script saphda_logic.groovy. The transformation derives the Analytics Cloud fact data payload from the SAP IBP for demand response.

- IBP on Demand is based on ISO 8601-1:2019 extended timestamp format (YYYY-MM-DDTHH:MM:SS), SAP Analytics Cloud model is based on Calendar Weeks (YYYYMM), so this mapping is done in the method
- The SAP Analytics Cloud dimensions members for SAP_ALL_COMPANY_CODE and SAP_ALL_SALESORGANISATION are derived from CUSTID

2.5 Writing into SAP Analytic Cloud

Data is written into SAC in chunks (by default month by month). These data chunks are written into SAC via a Job, which is opened with the first package and closed with the last. With the last data package, the job is validated and data is committed. Please refer to the documentation of the SAP Analytic Cloud API for further information.

2.6 Properties of the Integration Flow

All custom properties used in this Integration Flow are declared in the content modifier “initialize saphda properties”. But the property saphda_queryParameters needs to be refreshed in each loop, which happens in the content modifier “Refresh saphda properties”. So the odata filter for IBP needs to be customized identically in these two content modifiers.

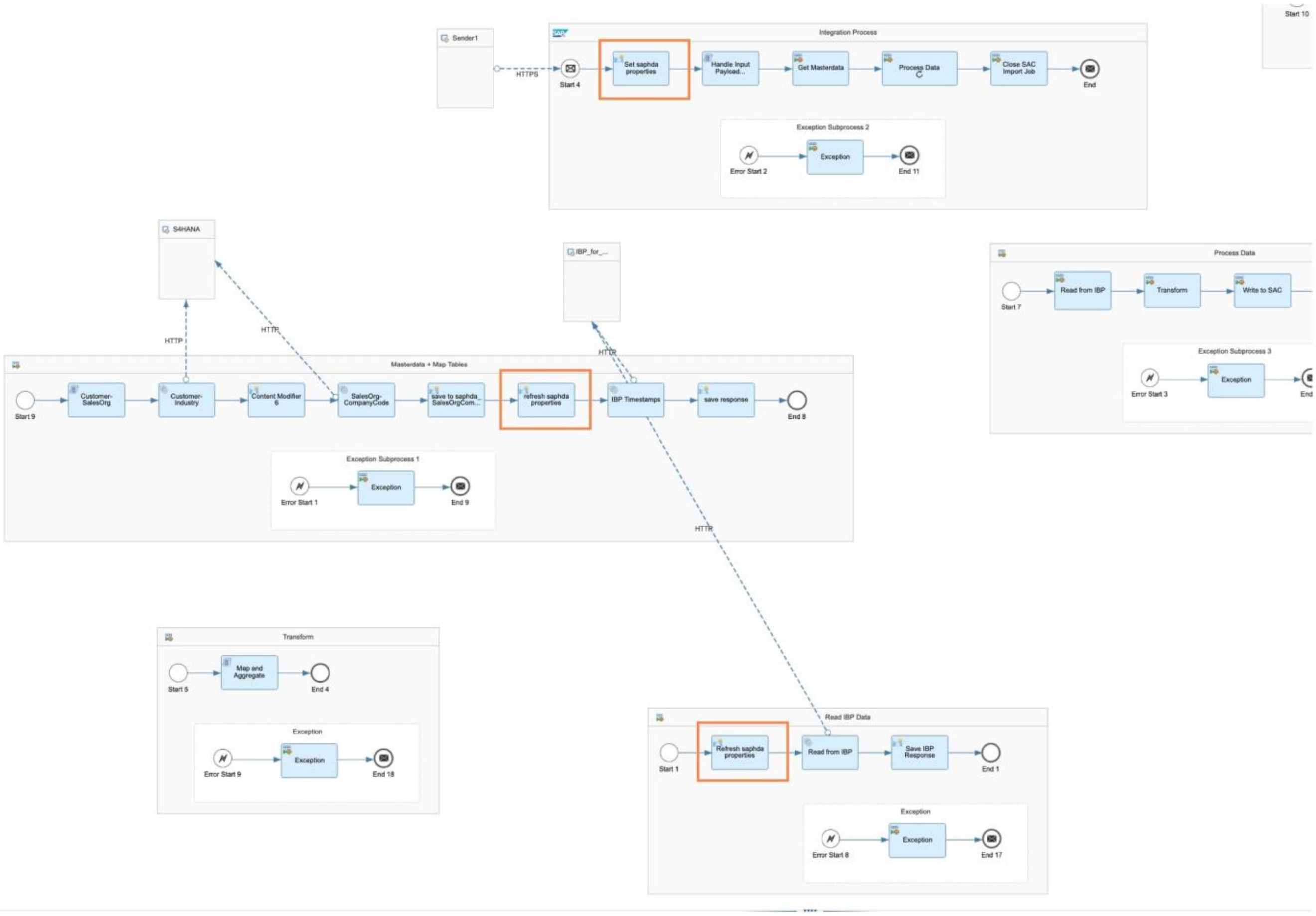


Figure 2 red highlighted are custom properties to be configured

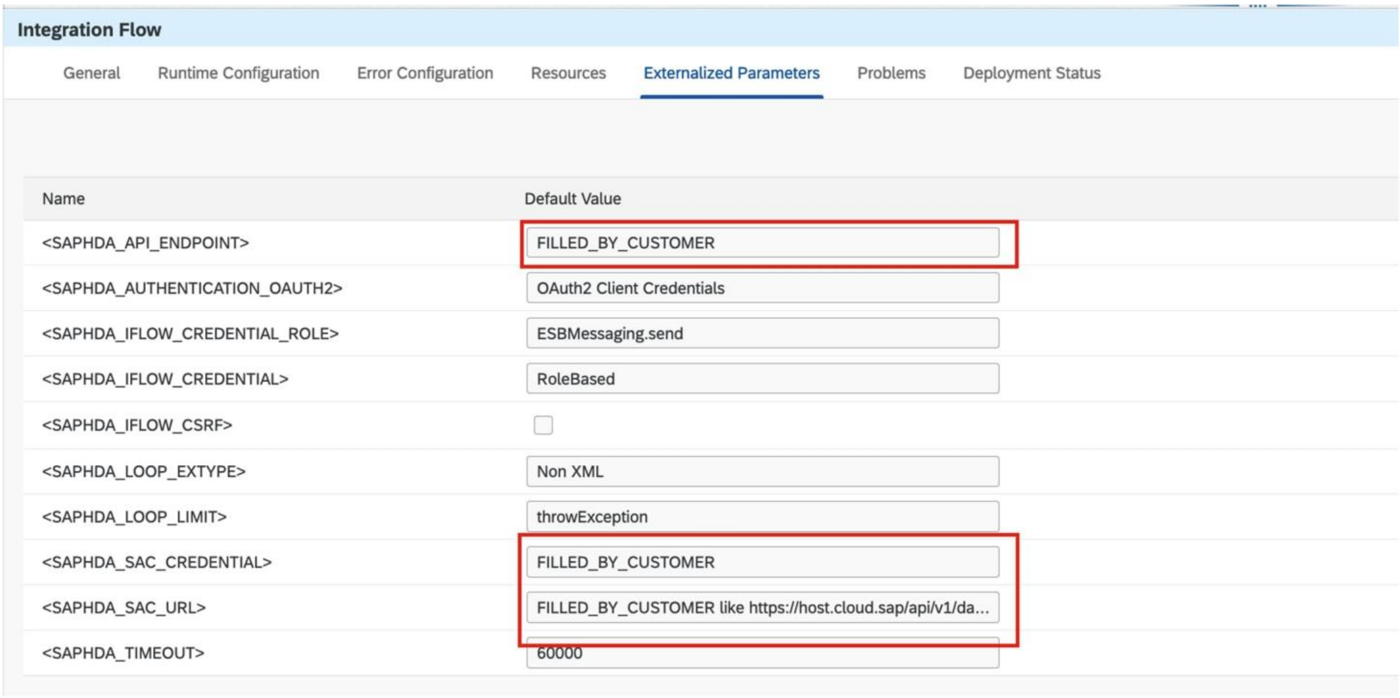
3 Configuration steps on SAP Cloud Integration

3.1 Configure Receiver Adapter

Receivers are connecting SAP Analytics Cloud, SAP IBP for Demand, SAP S/4HANA. If the flow should be used without adjustments, it is necessary to have the Content Packages Commercial Planning for SAP Analytics Cloud installed. In all systems user and authorizations needs to be granted. For using the master data mapping as they are built by delivery, also a S/4HANA system, user and access needs to be available. Please refer to the relevant documentation.

The following configuration is necessary

- Credential Artifacts
  - SAP Analytics Cloud with the credentials of an App Integration oAuth authorization.
    Externalized Parameter <SAPHDA_SAC_CREDENTIAL> ○ IBP for Demand with IBP External Planning Data
    Integration Communication Scenario SAP_COM_0720.
    Externalized Parameter <SAPHDA_IBP_CREDENTIAL> ○
    S/4HANA Scenario SAP_COM_0087
    Externalized Parameter <SAPHDA_S4_CREDENTIAL>
- URLs for the Systems ○ SAP Analytics Cloud
    Externalized Parameter <SAPHDA_SAC_URL>
    Example: https://host.cloud.sap/api/v1/dataimport/
    Be aware that the URL ends with a slash ○
    IBP for Demand
    Externalized Parameter <SAPHDA_IBP_URL>
    Example: https://host.com/sap/opu/odata/IBP/PLANNING_DATA_API_SRV/<planningareaname>
    Be aware that there is no slash at the end of the URL
  - S/4HANA
    Externalized Parameter <SAPHDA_S4_URL>
    Example: https://host.com/sap/opu/odata/sap/
    Be aware that the URL end with a slash



Name	Default Value
<SAPHDA_API_ENDPOINT>	FILLED_BY_CUSTOMER
<SAPHDA_AUTHENTICATION_OAUTH2>	OAuth2 Client Credentials
<SAPHDA_IFLOW_CREDENTIAL_ROLE>	ESBMessaging.send
<SAPHDA_IFLOW_CREDENTIAL>	RoleBased
<SAPHDA_IFLOW_CSRF>	□
<SAPHDA_LOOP_EXTYPE>	Non XML
<SAPHDA_LOOP_LIMIT>	throwException
<SAPHDA_SAC_CREDENTIAL>	FILLED_BY_CUSTOMER
<SAPHDA_SAC_URL>	FILLED_BY_CUSTOMER like https://host.cloud.sap/api/v1/da...
<SAPHDA_TIMEOUT>	60000

Figure 3 Configuration of the Externalized Parameter, like the SAC URL