Building block Configuration Guide
Commercial Planning Send SAP Analytics Cloud Marketing model drivers to IBP for Demand
May 2023
English

CUSTOMER

Commercial Planning Send SAP Analytics Cloud Marketing model drivers to IBP for Demand

Contents

1 Prerequisites

The Integration Flow “SAP Analytics Cloud Integration with SAP IBP for demand” connects the content package model for Marketing Planning in SAP Analytic Cloud with SAP IBP for Demand. This flow sends the (planned drivers) data from SAC into IBP. There is also an Integration Flow available to read (baseline quantities) data from IBP to send them into SAC.

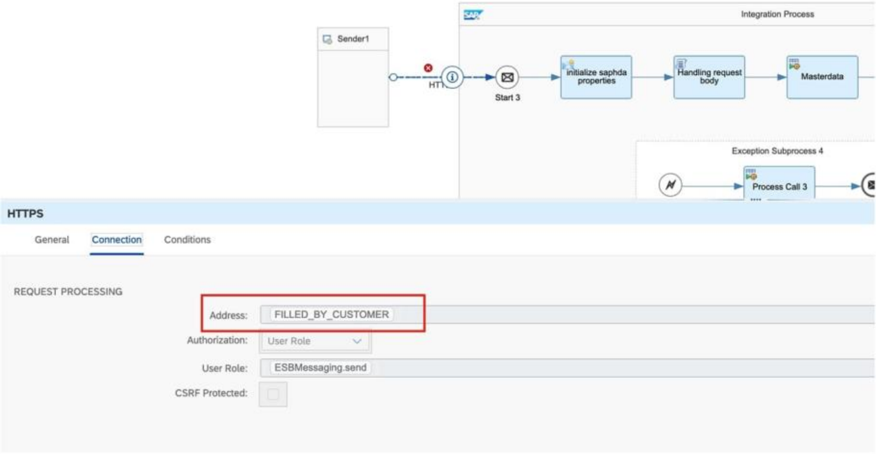
This Integration Flow is a possible implementation approach. But it is necessary to check the individual business needs.

2 Documentation

The flow reads fact data from SAP Analytics Cloud model, transforms the data, and writes the data into IBP for demand. 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 IBP month by month. As the data is aggregated, it is relevant to have all QUANITY 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 /write2ibp so that the Integration Flow can be called via the URL that can be found in the CI Monitor section.



HTTPS

General | Connection | Conditions

REQUEST PROCESSING

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). The date properties are arrays, as they should be filled by MultiAction parameters.

An example payload looks like:

```
{
   "modelID": "Cepk9k03peaigaeka98s7gqb32q",
   "calmonthFrom" : [["(all)","2024","202401"]],
   "calmonthTo" : [["(all)","2024","202412"]] }
```

2.2 Reading data from SAP Analytics Cloud

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

By delivery the configuration is set to

- Version = public.Plan
- SAP_FI_XPA_GLAccount = 65301000
- Date = is filled automatically by the function semanticPartition of groovy script saphda_logic.groovy

2.3 Transformation

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

- Aggregates the QUANTITY. The SAP Analytics Cloud API does not allow a selection query, so all properties are selected, also the ones not necessary to send to IBP. Over these properties the QUANTITY is aggregated
- The SAP Analytics Cloud model is based on Calendar Weeks (YYYYMM), IBP on Demand expects an ISO 8601-1:2019 extended timestamp format (YYYY-MM-DDTHH:MM:SS), so this mapping is done in the method as well - Country and Currency are added

By default, the following fields are sent to IBP

- Period Month: PERIODID3_TSTAMP
- Product: PRDID
- Customer Country: CUSTCOUNTRY
- Currency: CURRID
- Driver Quantity: MARKETINGBUDGETSAC

If different fields should be sent to SAP IBP for demand, also the method transform of groovy script saphda_logic.groovy needs to be adjusted.

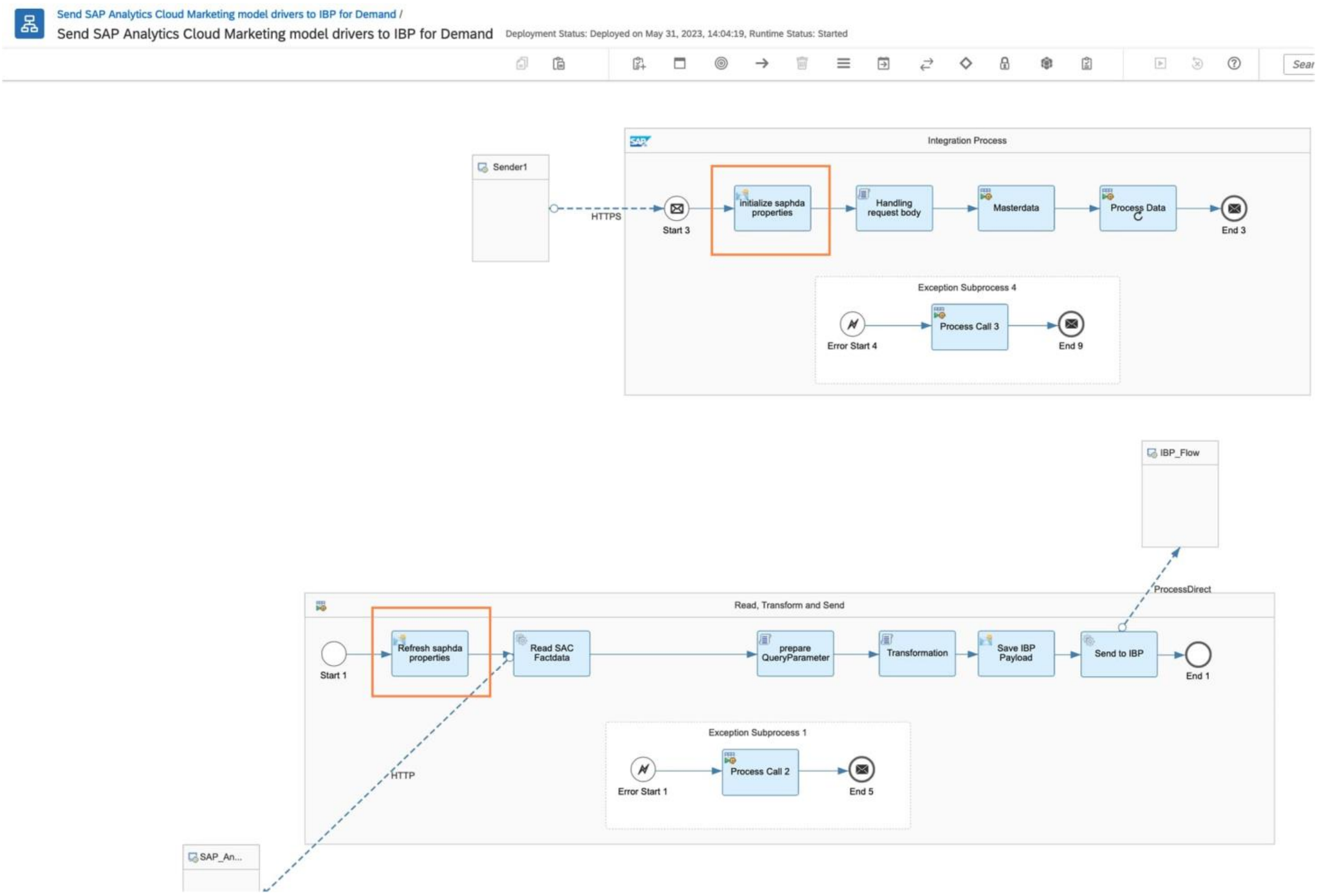
2.4 Writing into IBP

Writing into IBP is done as separate flow via process direct call. All customizing is done via exchange properties in this Marketing Integration Flow. Values are processed in the method transformation of groovy script saphda_logic.groovy.

- IBP hostname: saphda_ibp_url
  Example: https://my-ibp-api.host.ondemand.com
- credentialsName: saphda_ipb_credential
  The name of the security material, defined as basic authorization.
- planningArea: saphda_ibp_planningarea Example: XPACNT2305

2.5 Properties of the iflow

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 modifier.



Send SAP Analytics Cloud Marketing model drivers to IBP for Demand /

Send SAP Analytics Cloud Marketing model drivers to IBP for Demand Deployment Status: Deployed on May 31, 2023, 14:04:19, Runtime Status: Started

Figure 2 the odata filter reading data from SAC needs to be adjusted in the red highlighted boxes

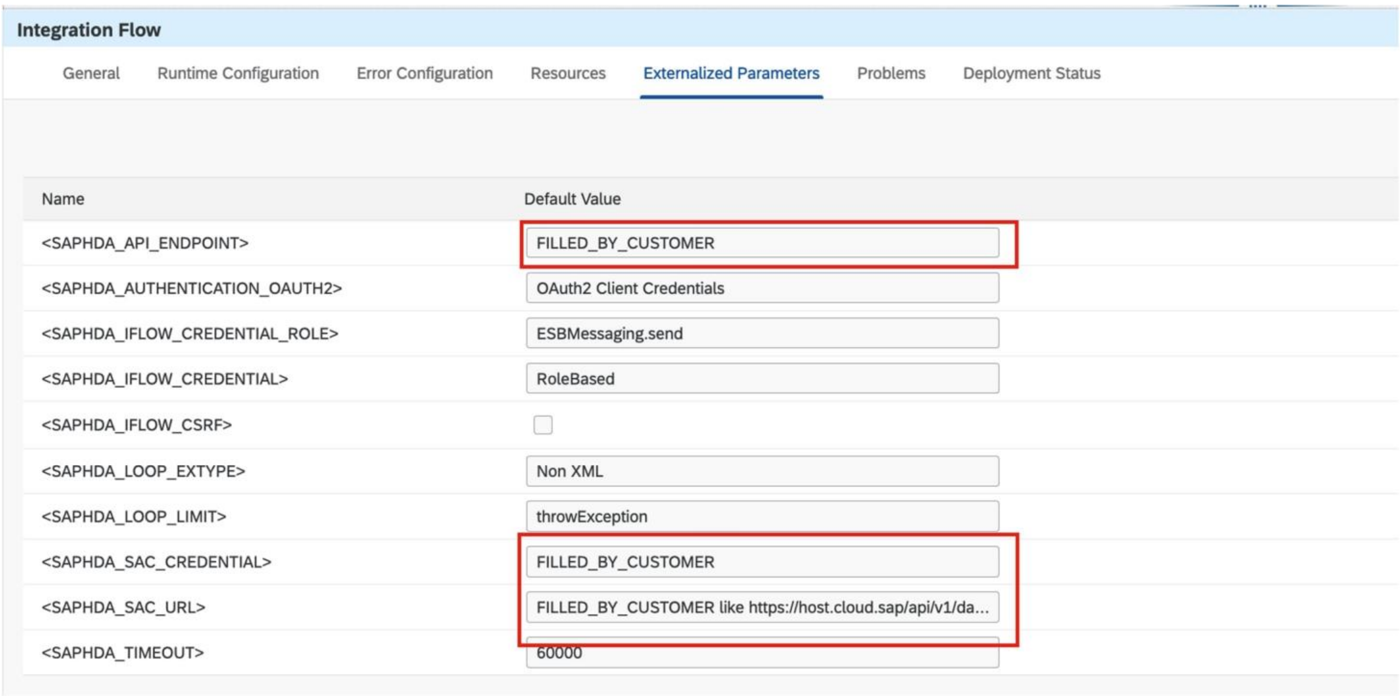
3 Configuration steps on SAP Cloud Integration

3.1 Configure Receiver Adapter

Receivers are connecting SAP Analytics Cloud and SAP IBP for Demand. If the flow should be used without adjustments, it is necessary to have the Content Packages for SAP Analytics Could installed, as well as IBP. In both systems user and authorizations needs to be granted. Please refer to the relevant documentation.

The following configuration is necessary.

- Two 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. Exchange Property: saphda_ibp_credential
- URLs for the Systems ○ SAP Analytics Cloud
  Externalized Parameter <SAPHDA_SAC_URL>
  Example: https://host.cloud.sap/api/v1/dataexport/providers/sac/
  Be aware that the URL ends with a slash ○
  IBP for Demand
  Exchange Property: saphda_ibp_url
  Example: https://host.ondemand.com
  Be aware that there is no slash at the end of the URL
- Planning Area for IBP on demand ○ Exchange Property:
  saphda_ibp_planningarea



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