# Content

<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>About This Document</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Integration Technologies</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Integration</td>
<td>8</td>
</tr>
<tr>
<td>3.1</td>
<td>Inbound</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Loyalty Data</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Market Data and Events</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Sales and Service Data</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Financial Data</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Industry Data</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Social Media, Commerce, and IoT</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Extensions</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Outbound</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Interaction Center</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Extensions</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Personalized Commerce</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>Mobile, Social, and Digital Channel</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>Supportive Integrations</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>Service Provider Integrations for Emails and Text Messages</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>SAP Consumer Insight 365</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>SAP HANA Rule Framework (HRF)</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>SAP Hybris customer journey manager add-on</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>SAP Hybris Digital Asset Management by OpenText</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>SAP Hybris Product Content Management (PCM)</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>SAP Visual Business and here.com</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Data Export and Data Import</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Upload and Download of Campaign Content for Editing in External HTML-Editors</td>
<td>56</td>
</tr>
<tr>
<td>4</td>
<td>APIs and OData Services</td>
<td>57</td>
</tr>
<tr>
<td>4.1</td>
<td>Channel Affinity</td>
<td>57</td>
</tr>
<tr>
<td>4.2</td>
<td>Import Business Partners Using Odata Service</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Technical Prerequisites</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Basic Concepts</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Structure of OData Service CUAN_BUSINESS_PARTNER_IMPORT_SRV</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>Field Extensibility</td>
<td>62</td>
</tr>
<tr>
<td>4.3</td>
<td>Generic OData Service for Offer Discovery</td>
<td>63</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>4.4</td>
<td>Importing Data of Campaigns and Target Groups Using an OData Service</td>
<td>68</td>
</tr>
<tr>
<td>4.5</td>
<td>Import of Offers Using an OData Service</td>
<td>70</td>
</tr>
<tr>
<td>4.6</td>
<td>Import of Campaign Execution Plans Using an OData Service</td>
<td>82</td>
</tr>
<tr>
<td>4.7</td>
<td>Import of Actual and Committed Spend Using an OData Service</td>
<td>85</td>
</tr>
<tr>
<td>4.8</td>
<td>Edit Messages Used in Campaigns in External Systems Using a Public OData Service</td>
<td>91</td>
</tr>
</tbody>
</table>
Document History

In the following you will find the major document changes for the guide. The newest changes are always on the top.

Table 1:

<table>
<thead>
<tr>
<th>Change Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 13, 2017 (Release to Customer)</td>
<td>Initial setup of the guide.</td>
</tr>
</tbody>
</table>
1 About This Document

This guide is the central starting point for the integrations with SAP Hybris Marketing.

Use this guide to get an overview of the available integrations with SAP Hybris Marketing. It should guide you through all the required steps for a successful integration with an external system.

You can find the most current information about the integrations with SAP Hybris Marketing on the SAP Help Portal under https://help.sap.com/mkt Integration. We strongly recommend that you use the documents available here. The guides are regularly updated.

Feedback

We’d really like to know what you think of the quality, structure or content of this guide. Please send your feedback to us at mailto:saphybrismarketingfeedback@sap.com.
# 2 Integration Technologies

Here you can find an overview about the integration technologies used in your solution.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP</td>
<td>SOAP is a protocol specification for exchanging structured information in the implementation of web services in computer networks. The message format is based on XML. Message transfer is based on other web protocols, usually HTTP(S).</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer (REST) is a architecture style for creating scalable web services. REST services are usually based on HTTP(S). They use HTTP URIs for resource identification and HTTP methods for service operations. It is used widely as an alternative to SOAP, as REST services usually provide better performance, scalability and simpler interfaces.</td>
</tr>
<tr>
<td>OData</td>
<td>OData provides a protocol for queryable and interoperable RESTful APIs. It provides an entity-based data model and a query language. Create, read, update and delete methods expressed using HTTP methods.</td>
</tr>
<tr>
<td>RFC</td>
<td>Call of a function module that runs in a different system (destination) from the calling program. Connections are possible between different AS ABAP and between an AS ABAP and a non-SAP system. In non-SAP systems, instead of function modules, special programmed functions are called, whose interface simulates a function module.</td>
</tr>
<tr>
<td>CSV</td>
<td>A comma-separated values (CSV) (also sometimes called character-separated values) file stores tabular data (numbers and text) in plain-text form. CSV files are widely used as import or export format and can be down- and uploaded to many systems.</td>
</tr>
<tr>
<td>ESP</td>
<td>SAP Event Stream Processor is a high performance complex event processing engine that collects and analyzes streams of events (messages) in real-time, as fast as they arrive. It can be used in conjunction with SAP HANA to collect streams of data, process the data, capture it in HANA, and monitor the data to generate alerts or immediate response.</td>
</tr>
<tr>
<td>SLT</td>
<td>System Landscape Transformation or short SLT is a tool for data migration and data replication: replicates tables in (near) real time from SAP systems to SAP HANA.</td>
</tr>
<tr>
<td>DS</td>
<td>Data Services (DS) provides a data integration platform to integrate and transform heterogeneous data (e.g. Web, RDBMS, XML, Flat files) using ETL processes.</td>
</tr>
<tr>
<td>Technology</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SAP MW</td>
<td>SAP CRM Middleware replicates, synchronizes and distributes data between different components of a SAP CRM solution. It links together the various types of data producers (such as ERP back end, SAP NetWeaver Business Intelligence, SAP APO, hybris Marketing and the CRM Server applications) to provide all participants with the information they require. Its main part is provided by the CRM Server.</td>
</tr>
<tr>
<td>HANA Co-Deployment</td>
<td>HANA co-deployment refers to multiple applications that are deployed to the same HANA instance. They can integrate by consuming HANA information models which integrate data of both applications.</td>
</tr>
</tbody>
</table>
3 Integration

Here you can find all existing integrations of SAP Hybris Marketing with external systems.

Overview

The graphic gives you an overview how the integrations roughly are working.

Inbound [page 8]
Outbound [page 32]
Supportive Integrations [page 43]

3.1 Inbound

Loyalty Data [page 9]
Market Data and Events [page 9]
Sales and Service Data [page 9]
Financial Data [page 14]
Industry Data [page 18]
Social Media, Commerce, and IoT [page 19]
Extensions [page 25]
3.1.1 Loyalty Data

SAP Hybris Marketing, loyalty option [page 9]
With this integration you can get interactions, contact scores, and offers.

3.1.1.1 SAP Hybris Marketing, loyalty option

With this integration you can get interactions, contact scores, and offers.

For more information, see the SAP Help Portal under https://help.sap.com/mkt ➤ SAP Hybris Marketing On Premise ➤ SAP Hybris Marketing Product Page ➤ Integration ➤ Integrating SAP Hybris Loyalty in SAP Hybris Marketing ➤

3.1.2 Market Data and Events

SAP XM [page 9]
With this integration you can push ad campaigns and get cookie-based user data and interactions from SAP Exchange Media (XM) for further analysis.

3.1.2.1 SAP XM

With this integration you can push ad campaigns and get cookie-based user data and interactions from SAP Exchange Media (XM) for further analysis.

For more information, see on SAP Help Portal under https://help.sap.com/mkt ➤ SAP Hybris Marketing On Premise ➤ SAP Hybris Marketing Product Page ➤ Integration ➤ Integrating SAP Hybris Marketing with SAP Exchange Media ➤

3.1.3 Sales and Service Data

SAP Customer Relationship Management (CRM) [page 10]
With the integrations for SAP CRM you can, for example, import business documents from SAP CRM Interaction Center from a system of SAP CRM.

SAP Hybris Cloud for Customer [page 11]
With these integrations you can transfer call activities using leads and you can get data, such as contacts, corporate accounts, leads, opportunities, call activities, appointments, sales tasks, phone calls, appointments, and visits from the connected systems.

**Presales / Sales [page 11]**
Set up the integration of a Sales system with SAP Hybris Marketing

**Importing Data for Customer Value Intelligence [page 12]**
With the integration for Customer Value Intelligence you can import data, such as sales orders, to your system from, for example, SAP Hybris Cloud for Customer.

**SAP Hybris Commerce [page 13]**
With these integrations you can connect with a system of SAP Hybris Commerce.

### 3.1.3.1 SAP Customer Relationship Management (CRM)

With the integrations for SAP CRM you can, for example, import business documents from SAP CRM Interaction Center from a system of SAP CRM.

You can integrate the following data from a system of SAP CRM with your system:

- Sales Business Documents such as Orders and Opportunities
- Call Lists and Call Activities to the SAP CRM Interaction Center


Note that you can use the following backend transactions for an automatic check of the required settings:

- In the SAP Hybris Marketing system:
  - SAP Hybris Marketing CRM Integration Check Report (CUAN_CRM_CHECK)
  - Condition Check (CUAN_TC_CHECK) in the folder Cuan CRM Integration Check.
- In the SAP CRM system: CRM_CUAN_CHECK: Consider [2202431](https://help.sap.com/mkt) to enable the check.

### SAP CRM Interaction Center

You can connect your system with the SAP CRM Interaction Center to exchange call lists, call activities, and recommendations and also import business documents.

**Recommendations**

Business Documents Generated By the Interaction Center


3.1.3.2 SAP Hybris Cloud for Customer

With these integrations you can transfer call activities using leads and you can get data, such as contacts, corporate accounts, leads, opportunities, call activities, appointments, sales tasks, phone calls, appointments, and visits from the connected systems.


3.1.3.3 Presales / Sales

Set up the integration of a Sales system with SAP Hybris Marketing

SAP Hybris Marketing can be integrated with SAP Hybris Cloud for Customer via SAP HANA Cloud Integration (SAP HCI), or SAP NetWeaver Process Integration SAP (PI).

Related Information

Integration with SAP Hybris Cloud for Customer [page 11]

3.1.3.3.1 Integration with SAP Hybris Cloud for Customer

Data exchange between Marketing and Sales.

By integrating SAP Hybris Marketing, and SAP Hybris Cloud for Customer via SAP HCI, or SAP PI, exchange of data between marketing and sales is enabled. This process ensures that Marketing provides Sales with highly qualified data that supports the conversion of potential buyers to real buyers.

3.1.3.4 Importing Data for Customer Value Intelligence

With the integration for Customer Value Intelligence you can import data, such as sales orders, to your system from, for example, SAP Hybris Cloud for Customer.

Import Business Documents Such As Sales Orders


- Deployment Scenarios: Deployment Scenarios B, C, and D
- ‹ Data Replication › Installation of SAP Landscape Transformation Replication Server
- ‹ Configuring Insight › Importing Data from External Source to “Margin Decomposition”
- ‹ Appendix › Replicated Tables in SAP Hybris Marketing

3.1.3.4.1 Integration with SAP ERP for Spend Planning

Prerequisites

You have maintained the following activities in Customizing for SAP Hybris Marketing under › Planning › Spend Management:

- Assign Marketing Area to Project Profile
- Maintain Project and WBS Element Prefix for ERP
- BAdl: Outbound System Integration for Cost References

You have maintained the following activities in Customizing for SAP ERP under › Project System:

- ‹ Costs › Activate Project Management in Controlling Area
- ‹ Structures › Operative Structures › Work Breakdown Structure (WBS) › Create Project Profile
- ‹ Structures › Templates › Standard Work Breakdown Structure › Settings for Standard and Operative WBSs › Maintain Project Types for Standard WBS Elements

You have maintained the SAP Landscape Transformation (SLT) replication so that the data will be replicated from SAP ERP into SAP Hybris Marketing.
Process

If you set up everything correctly, the following process should work:

1. You create a campaign in SAP Hybris Marketing with a marketing area and a target group, and an optional assignment to a program with a specific proposed spend.
2. In the Detailed Campaign Spend application, you assign one or more spend items with a planned amount.
3. After you released the campaign in SAP Hybris Marketing, the following objects are created in the project system of the connected SAP ERP:
   ○ One project representing the campaign
   ○ One or more work breakdown structure (WBS) elements representing the marketing spend

   **Note**
   ○ As soon as a campaign is released, the spend type of a spend item cannot be changed and a spend item cannot be deleted.
   ○ You can add spend items to a released campaign.

4. Then you perform the procurement where you post the costs on the corresponding WBS elements and the SAP ERP system processes and stores the cost-relevant data.
5. The data then will be replicated to SAP Hybris Marketing using the System Landscape Transformation (SLT).
6. In SAP Hybris Marketing the data is retrieved using SAP HANA views and shown as actual results in the Detailed Campaign Spend application.

### 3.1.3.5 SAP Hybris Commerce

With these integrations you can connect with a system of SAP Hybris Commerce.

For more information about the integration with SAP Hybris commerce, see the SAP Hybris Help Portal under [https://help.hybris.com](https://help.hybris.com) Integrations and Data Management SAP Integrations SAP Hybris Marketing Integration. There you will find the following integration scenarios that are delivered with SAP Hybris Commerce:

- SAP Hybris Marketing Recommendation
- SAP Hybris Marketing Segmentation (personalized content)
- Data Feeds for SAP Hybris Marketing (consumer- and sales-related business documents)

### Clickstreams

For more information about the integration of clickstreams, see Clickstream Event Capturing for SAP S/4HANA Marketing Cloud and SAP Hybris Marketing.
Offer Recommendation


Offer Discovery (Offer according to eligibility and validity)

For more information, see Generic OData Service for Offer Discovery [page 63].

3.1.4 Financial Data

Spend Integration with SAP ERP [page 14]

3.1.4.1 Spend Integration with SAP ERP

Importing Actual and Committed Spend from SAP ERP [page 14]

You can import actual and committed spend associated to campaigns from SAP ERP and make it available in the Detailed Campaign Spend application.

Exporting Spend Data to SAP ERP [page 15]

With the following instructions you are able to export the spend data from your system to SAP ERP. A campaign is represented by a project and a spend item as a WBS element. You use these WBS elements as account reference for further processing within SAP ERP.

3.1.4.1.1 Importing Actual and Committed Spend from SAP ERP

You can import actual and committed spend associated to campaigns from SAP ERP and make it available in the Detailed Campaign Spend application.

Prerequisites

- You have enabled the export of spend data to SAP ERP. For more information, see Exporting Spend Data to SAP ERP [page 15].
You have set up the SAP Hybris Marketing Cloud - SAP ERP Actual and Committed Spend integration package in SAP HANA Cloud Integration at https://cloudintegration.hana.ondemand.com/.

You have configured the communication scenario Marketing - Business Data Integration in SAP Hybris Marketing Cloud.

You have installed the latest version of the SAP CLOUD CUST ERP INTEGR 2.0 product in SAP ERP that contains the COD_ERP_INT 600 component. For information about this product and the CODERINT 600 add-on, see the corresponding documentation on SAP Service Marketplace at https://support.sap.com/software/patches/a-z-index.html C SAP CLOUD CUST ERP INTEGR 2.0.

You have the authorization to run the CUAN_ERP_MSM_EXTRACT_ACTUAL report in SAP ERP.

Context

You can import actual and committed spend data from SAP ERP using this integration.

Procedure

To import the actual and committed spend, run the CUAN_ERP_MSM_EXTRACT_ACTUAL report.

Results

The actual and committed spend associated to campaigns are shown in the Detailed Campaign Spend application.

3.1.4.1.2 Exporting Spend Data to SAP ERP

With the following instructions you are able to export the spend data from your system to SAP ERP. A campaign is represented by a project and a spend item as a WBS element. You use these WBS elements as account reference for further processing within SAP ERP.

Prerequisites

- You have SAP ERP 6.0 EHP 4 or higher.
- You have set up secure communication between SAP Hybris Marketing Cloud and SAP ERP. This is required so that SAP ERP gets secure requests from the Internet. One way of doing this is to configure a reverse proxy. For more information, enter the keywords Configuring Reverse Proxy on SAP Help Portal at http://help.sap.com.
Setting Up the Communication with SAP ERP

To set up the communication between SAP Hybris Marketing Cloud and SAP ERP, perform the following steps:

1. Create the system for outbound communication using the Communication Systems self-service configuration application. Enter the following data:

<table>
<thead>
<tr>
<th>System Name</th>
<th>Name of the SAP ERP system, for example, ABC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Name</td>
<td>Host address of the SAP ERP system, for example, ldiabc.corp.com.</td>
</tr>
<tr>
<td>Authentication Method</td>
<td>Select the certificate-based authentication.</td>
</tr>
</tbody>
</table>

2. Create a communication arrangement with a certificate-based authentication for the outbound scenario.

3. Create a communication arrangement for the Marketing - Planning Spend Integration (SAP_COM_0018) scenario using the Communication Arrangements self-service configuration application. Enter the following data:

<table>
<thead>
<tr>
<th>Common Data</th>
<th>Arrangement Name</th>
<th>SAP_COM_0018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Data</td>
<td>Communication System</td>
<td>The SAP ERP system created in the Communication Systems self-service configuration application in the previous step.</td>
</tr>
<tr>
<td>Outbound Communication</td>
<td>User Name/Certificate</td>
<td>Select certificate-based authentication for the outbound scenario. Download the certificate. It will be used later while setting up SAP ERP and configuring the security.</td>
</tr>
<tr>
<td>Outbound Services</td>
<td>Port</td>
<td>Port for the communication.</td>
</tr>
</tbody>
</table>
You must set up the following SOAP services in the order defined below and specify the path for each of them:

1. **Create service**, for example, `/sap/bc/srt/xip.sap/ecc_projecterpcrtrc1/<client of SAP ERP system>/<name of service/binding that will be used when configuring Web Services>/ <name of service/binding that will be used when configuring Web Services>`

2. **Update service**, for example, `/sap/bc/srt/xip.sap/ecc_projectupdrc/<client of SAP ERP system>/<name of service/binding that will be used when configuring Web Services>/ <name of service/binding that will be used when configuring Web Services>`

3. **Get service**, for example, `/sap/bc/srt/xip.sap/ecc_projecterpidqr1/<client of SAP ERP system>/<name of service/binding that will be used when configuring Web Services>/ <name of service/binding that will be used when configuring Web Services>`

The path for each service is defined during the SAP ERP setup.

For example, `<client of SAP ERP system>` could be 100, `<name of service/binding that will be used when configuring Web Services>` could be `cuan_msm`. The name you define in this step must be the same as the name defined during the SAP ERP setup.


### Setting Up SAP ERP

1. Configure the Web service runtime as target system connection. For more information, enter the keywords **Configuring the Web Service Runtime** on SAP Help Portal at [http://help.sap.com](http://help.sap.com).

2. Configure your security settings for the service provider and service consumer using the SOA Manager. For more information, enter the keywords **Secure Runtime Configuration with the SOA Manager** on SAP Help Portal at [http://help.sap.com](http://help.sap.com).

3. Configure service definitions in the Web service configuration transaction (SOAMANAGER) for the following:
   - `ProjectERPCreateRequestConfirmation_In_V1(ECC_PROJECTERPCRTRC1)`
   - `ProjectERPUpdateRequestConfirmation_In(ECC_PROJECTUPDRC)`
   - `ProjectERPByIDQueryResponse_In_V1(ECC_PROJECTERPIDQR1)`

---

**Integration Guide 1702**

**Integration**
Configure the services as follows:

1. In the **Service and Binding Name** step, enter the same name in the **Service Name** and **New Binding Name** fields. This must be the same name as the one defined in step 3 of the communication setup with SAP ERP under [Outbound Services > Path].

2. In the **Provider Security** step, under **Transport Level Security**, select the **SSL (https)** radio button, and under **Transport Channel Authentication**, select **X.509 SSL Client Certificate**.

3. Go through the other steps without specifying any values and complete the configuration.

For more information, enter the keywords **Configuring a Service Provider** on SAP Help Portal at [http://help.sap.com](http://help.sap.com).

As in SAP Hybris Marketing Cloud, prefix **1_CUAN_MSM_<CampaignID>** is given for creation of projects and **1/<CampaignID>** for WBS elements in SAP ERP, no predefined coding mask is required for project coding key 1.

Project profile **CUAN01** has to be configured in SAP ERP.

### 3.1.5 Industry Data

#### SAP Billing and Revenue Innovation Management (BRIM) [page 18]

With this integration you can get business support system (BSS) business documents from the connected SAP BRIM such as FICA Billing.

#### SAP Convergent Mediation [page 19]

With this integration you can get OSS data from SAP Convergent Mediation using a digital root connector (Partner Solution from DigitalRoute).

#### SAP Promotion Management [page 19]

With this integration you can import offers from SAP Promotion Management in your system.

#### SAP CAR and SAP Retail [page 19]

With this integration you can get POS data from a connected system of SAP CAR or SAP Retail.

### 3.1.5.1 SAP Billing and Revenue Innovation Management (BRIM)

With this integration you can get business support system (BSS) business documents from the connected SAP BRIM such as FICA Billing.

For more information, see SAP Service Marketplace [http://service.sap.com/rds-ymkt](http://service.sap.com/rds-ymkt) and search for **SAP Hybris Marketing for Telecommunications**.

3.1.5.2 SAP Convergent Mediation

With this integration you can get OSS data from SAP Convergent Mediation using a digital root connector (Partner Solution from DigitalRoute).

For more information, see the setup guide for SAP Convergent Mediation by DigitalRoute Application Operation Integration (Solution Manager 7.1) (http://www.sap.com/docs/download/2016/12/c21c4103-9c7c-0010-82c7-eda71af511fa.pdf).

3.1.5.3 SAP Promotion Management

With this integration you can import offers from SAP Promotion Management in your system.

For more information, see Import of Offers Using an OData Service [page 70] or also on the SAP API Business Hub https://api.sap.com/SAP Hybris Marketing Cloud Import Offer.

3.1.5.4 SAP CAR and SAP Retail

With this integration you can get POS data from a connected system of SAP CAR or SAP Retail.

For more information, see SAP Service Marketplace http://service.sap.com/rds-cei and choose SAP Hybris Marketing for Retail.

3.1.6 Social Media, Commerce, and IoT

- SAP Hybris Conversion [page 20]
  With this integration you can import clickstreams from SAP Hybris Conversion

- SAP Hybris Profile [page 20]
  With this integration you can import user profiles and interactions from SAP Hybris Profile in your system and use YaaS-based web tracking.

- SAP JAM [page 20]
  With the integrations of SAP Jam you can get user profiles and product reviews, as well as use the collaboration for marketing planning, campaigns, and sentiment engagement.

- Facebook Pages, Google+, and Twitter [page 21]
  With this integration you can exchange social posts with Facebook Pages, Google+, and Twitter using sentiment engagement.

- Facebook and Instagram [page 21]
  With this integration you can plan and create campaigns that push ads to Facebook and in addition also to Instagram. From Facebook then you get actual spend and campaign success data that you can analyze in your system.
The integration with Gigya provides user profiles, marketing attributes, and permissions from the Gigya ID management.

3.1.6.1 SAP Hybris Conversion

With this integration you can import clickstreams from SAP Hybris Conversion.

For more information, see the SAP Help Portal under [SAP Hybris Marketing On Premise > SAP Hybris Marketing Product Page > Integration > Data Management Upload Interfaces > Import Using Web Service > Importing Products and Product Categories Using Web Service].

3.1.6.2 SAP Hybris Profile

With this integration you can import user profiles and interactions from SAP Hybris Profile in your system and use YaaS-based web tracking.

For more information, see the SAP Help Portal under [SAP Hybris Marketing On Premise > SAP Hybris Marketing Product Page > Integration > Integrating with SAP Profile Services for Marketing].

3.1.6.3 SAP JAM

With the integrations of SAP Jam you can get user profiles and product reviews, as well as use the collaboration for marketing planning, campaigns, and sentiment engagement.

SAP Jam Communities provide community functionalities that tightly integrate with SAP Hybris Marketing to drive engagement, instill buying confidence and increase loyalty through the delivery of optimized content during the customer journey, e.g. to read and write product reviews. These give valuable insight on the context of a consumer. Interactions on the community enrich the consumer profile in SAP Hybris Marketing and allow for specific marketing activities.

For a detailed description about how to configure the SAP S/4HANA to SAP Jam Integration, see [https://rapid.sap.com/bp/#/browse/scopeitemversions/74ac0570703741f588617e8f7508a393].

For more information, see SAP Help Portal [http://help.sap.com/mkt > SAP Hybris Marketing On Premise > SAP Hybris Marketing Product Page > Installation and Upgrade > Installation Guide] and then the following sub-chapters:

- Setting up SAP Jam Integration with Post Groups
- SAP Jam Integration with My Marketing Budget
- Setting up SAP Jam Integration with Campaigns (Optional)
- Configuring Campaigns > Customizing
3.1.6.4 Facebook Pages, Google+, and Twitter

With this integration you can exchange social posts with Facebook Pages, Google+, and Twitter using sentiment engagement.

For more information, see:
- SAP Service Marketplace [http://service.sap.com/mkt](http://service.sap.com/mkt) choose the required release and look for Data Management Upload Interfaces, sub-chapter Import Using OData Service CUAN_IMPORT_SRV

3.1.6.5 Facebook and Instagram

With this integration you can plan and create campaigns that push ads to Facebook and in addition also to Instagram. From Facebook then you get actual spend and campaign success data that you can analyze in your system.

Prerequisites on Facebook

Before you begin, a few things need to be done:

- You need your own Facebook app that must be reviewed and released for productive usage by Facebook. When starting the review process, mention that you are using SAP Hybris Marketing.

  **Note**

  Note that a prerequisite for the approval is a link to a data privacy policy that is visible to every user of the app. Ensure that your company has such a policy in place.

- Look up the application ID (*App ID*) and client secret (*App Secret*) in Facebook for later use when configuring the OAuth 2.0 Client.

- To actually do advertising on Facebook you need a Facebook ad account. It is recommended to use Facebook Business Manager. For details refer to the Facebook documentation. If you work together with a marketing agency, you have to clarify who owns and manages the ad account. In any case your users need marketer permissions on the ad account. You can also work with multiple ad accounts (such as one account per marketing area).
Back-End Configuration

i  Note

Ensure the required role: SAP_CUSTOMER_ANALYTICS_ADMIN (single) is included in composite role SAP_CEI_TECHNICAL_CONF.

1. Configure an OAuth 2.0 client:
   ○ From the SAP Easy Access menu, start the transaction OA2C_CONFIG.
     This will open the OAuth 2.0 Client Configuration in a Web browser. Alternatively, you can open the URL with:
   ○ From the OAuth 2.0 Clients window, choose Create, select FACEBOOK from the client profile drop-down menu and enter the client identification (App ID) number you received during registration of the application with Facebook.
   ○ Enter the following scenario-specific Target Endpoint:
     /sap/bc/ui5_ui5/sap/CUAN_NAV_TO/index.html#CUAN_TI_F_INI_MKT_AUTOMATION_ENDPOINT.
     The target endpoint entered on your ABAP application server is used to redirect the browser of the end user after completing the authorization process.
   ○ Under General Settings, enter the Client Secret (App Secret) information you received during registration of the application with Facebook and press Enter to confirm the entry.
   ○ From the Scopes tab page, verify that FACEBOOK is displayed in the profile table, and Ads Management and Public Profile are listed in the Scopes table.
   ○ Save the OAuth 2.0 Client configuration.

2. Maintain Facebook application settings:
   ○ As in step 1 from the OAuth 2.0 Client (transaction OA2C_CONFIG) window, copy the Redirection URI into the clipboard.
   ○ Add it to Facebook under App settings > Advanced > Valid OAuth redirect URIs. This will make your OAuth 2.0 client known to the OAuth 2.0 authorization server on Facebook.

3. Configure proxy settings:
   Access to the internet from the AS ABAP system is made possible through a proxy server. If not yet configured, please configure your proxy server settings.
   Although proxy settings may be active, the HTTPS protocol might not have a proxy listed. If so, add your proxy.

4. Configure Secure Socket Layer (SSL) settings:
   ○ Go to https://www.facebook.com and click on the lock symbol next to the URL to get the site information.
○ Locate the DigiCert High Assurance EV Root CA and save it as a file.
○ Start transaction STRUST in SAP Easy Access menu and switch to change mode.
○ Choose PSE SSL Client Anonymous.
○ From the Certificate area, choose Import Certificate and import the saved Facebook certificate file.
○ Choose Add to Certificate List and save the PSE file.

The AS ABAP will now trust SSL servers whose identity is confirmed by this certificate.

5. Activate the FB campaign category in Customizing from Define Campaign Categories ➔ Assign Actions to Categories.

3.1.6.5.1 Integration With Facebook

Use

With the outbound integration with Facebook, you can leverage the data and features of SAP Hybris Marketing together with the predictive capabilities of SAP HANA to address the right people with ads on Facebook. You can create and track the success and actual spend of your Facebook campaigns or define a custom audience, the Facebook equivalent of target groups, for use in Facebook ads.

Prerequisites

Before you can begin creating a campaign in Facebook:

● You must have created your own app in Facebook and have a company data privacy policy that is visible to every user of the app.

   Note

   Your app will be reviewed by Facebook first before it can be used officially. Please ensure you have a link to your companies data privacy policy before you submit your app for review.

● You need a Facebook ad account and a user that has been assigned either the Ad Account Admin or Ad Account Advertiser permissions for at least one ad account in Facebook. If you work together with a marketing agency you have to clarify who owns and manages this account.

Facebook also requires a check for marketing permissions when using their custom audiences, a check that is done by default in SAP Hybris Marketing. In SAP Hybris Marketing, if the user has not given their permission for their data to be used for advertising purposes it can not be used in campaigns created either for Facebook custom audiences or third-parties. However, some countries have implicit opt-in permission. This means that if the user does not specifically forbid SAP Hybris Marketing from using their user for advertising purposes, the user information can be transferred to a custom audience in Facebook.

Please refer to for more information.

Your marketing ad account manager and system administrators can assist you with any questions about these prerequisites.
More Information

For information about setting up the system, see the installation and upgrade guides for SAP Hybris Marketing at http://help.sap.com/mkt.

For more information about Customizing settings, see Customizing for SAP Hybris Marketing in the system.

### 3.1.6.6 Gigya ID Management

The integration with Gigya provides user profiles, marketing attributes, and permissions from the Gigya ID management.

#### Overview

See the following diagram for an overview of the integration based on traditional registration tools:

![Diagram of traditional registration integration](image)

See the following diagram for an overview of the integration based on social registration tools:

![Diagram of social registration integration](image)
User Data

The integration scenario processes the user data as follows:

Table 5: User Data

<table>
<thead>
<tr>
<th>Data Source (Gigya)</th>
<th>Target (SAP Hybris Marketing)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gigya ID</td>
<td>ID</td>
<td>Becomes own facet with ID_ORIGIN = GIGYA_ID</td>
</tr>
<tr>
<td>Social User ID</td>
<td>ID</td>
<td></td>
</tr>
<tr>
<td>Email address</td>
<td>ID</td>
<td>For the facet with ID_ORIGIN = EMAIL</td>
</tr>
<tr>
<td>Name, address, demographic attributes</td>
<td>Standard attributes of interaction contact</td>
<td>Written into facet data table, and from there propagated into the golden record</td>
</tr>
</tbody>
</table>

3.1.7 Extensions

Customer Web Sites and Landing Pages [page 25]

With this integration you can get marketing permissions and contact data from your web hosting service and the used content management system (CMS).

3.1.7.1 Customer Web Sites and Landing Pages

With this integration you can get marketing permissions and contact data from your web hosting service and the used content management system (CMS).

For more information, see Setting up Landing Pages [page 25].

3.1.7.1.1 Setting up Landing Pages

In Acquisition, you are provided with the Landing Pages application. This app allows you to design landing pages to collect interaction contact and marketing permission data. The content of the app is also included in the Content Studio application.

For security and performance reasons, you must deploy the landing pages that you have created using this app, on your Web server available to make them available to the internet. The Web server must send the collected data to the SAP Hybris Marketing server using the OData service CUAN_CONTENT_PAGE_RESULT_SRV. The service saves the data and triggers follow-on actions.
Implementation of the landing pages includes the following parts:

- Deploying the files on your Web server
- Implementing the Web server in order to forward the results to the SAP Hybris Marketing system.

Assign Users to Roles

A marketing expert can only use Landing Pages, if his or her user is assigned to your copy of the following roles:

- SAP_CEI_CPM_FLP (business catalog role; no copy required - you can assign the delivered standard role)
- SAP_CEI_CONTENT_PAGES (back-end role; included in composite role SAP_MARKETING_CAMPAIGNS)
- SAP_CEI_HOME (for personalization purposes)

Configuration steps related to the SAP Hybris Marketing server require a user which has your copy of the following role assigned:

- SAP_CEI_CONTENT_PAGE_RSLT (back-end role; included in composite role SAP_MARKETING_CAMPAIGNS)

**i Note**

Your copy of the latter role is also required for testing newly created landing pages.

3.1.7.1.1.1 Deploying the Landing Page

The source code (HTML) for every landing page you design must be downloaded via the user interface and deployed on your Web server.

In addition, you can download a style sheet (CSS) and a JavaScript file. You must adapt the JavaScript file according to your requirements and your system setup. This step is required for your initial system setup and allows you to deploy the CSS and JavaScript files. You do not need to adapt these files again until you perform an upgrade to a new release. Adapting the files when you upgrade ensures that you can avail of new features.

By default, the HTML file tries to load both files with the names sapContentPage.css and sapContentPage.js from the same folder in which the HTML file is located. If you want to adjust those names or the file paths you must change the HTML file content.

3.1.7.1.1.2 Adjusting the JavaScript File

After downloading the JavaScript file you must adjust the Web service path written in the file.

The base is the path for the OData service on the SAP Hybris Marketing server:

- /sap/opu/odata/sap/CUAN_CONTENT_PAGE_RESULT_SRV

You must adjust this path according to the Web server implementation (see section Implementing the Result OData Service [page 27]).
When a user opens a landing page, the path is loaded via an HTTP HEAD request in order to fetch a CSRF token. Further data requests are sent to the result path which is added to the base path. The result path can be adjusted or left empty in the JavaScript file. The default path `ResultHeaders` describes the OData service entity used for the results.

You are not required to perform any implementation for the front end. The JavaScript that is delivered is capable of collecting the user input independently. It is possible to adapt and enhance the landing page HTML file that is generated, but you must ensure you preserve the integrity of the standard structure.

### 3.1.7.1.1.3 Implementing the Result OData Service

You must implement your Web server in a way that makes it capable of receiving the requests of the landing page JavaScript files and forwarding them to the SAP Hybris Marketing server. By default, the requests are ready for the result OData service `CUAN_CONTENT_PAGE_RESULT_SRV` and do not need to be adjusted.

**i** Note

In some usage scenarios, the request data must be enhanced to enable all features. To do this, you must decode the JSON payload string and add the appropriate attributes before encoding the JSON string again for the OData service.

The actual implementation depends on the technology and development language that you use in your company. See an example implementation in PHP below.

**Caution**

The following code is an example implementation. SAP does not take responsibility if you use it in your productive system.

**Example**

```php
<?php
/**
 * This class is an example implementation
 * of a PHP based landing page integration.
 * Note: This is a template, which is used at your own risk.
 */
class LandingPageIntegration
{
    /**
     * The BasePath is the URL for the system
     * including the landing page result OData service.
     */
    const BASE_PATH = "https://<server>:<port>/sap/opu/odata/sap/
    CUAN_CONTENT_PAGE_RESULT_SRV/";
}/**
```
* The ResultHeadersPath is the name of the ResultHeaders entity which is used for processing the landing page results.
* 
* @var string
*/
const RESULT_HEADERS_PATH = "ResultHeaders";
/**
 * The credentials are used for authenticating on the system.
 * This is usually a dedicated system or communication user with the integration role assigned.
* @var string
*/
const CREDENTIALS = "USERNAME:PASSWORD";
/**
 * The cookies are remembered between consecutive OData requests to implement the session handling and security measures of the SAP Gateway.
* @var string
*/
private $cookies = "";
/**
 * The CSRF-Token is required for the OData service communication and must be fetched before it is possible to perform any changing requests such as 'POST'.
* @var string
*/
private $csrfToken = null;
/**
 * This method is the main entry point for processing the requests received from landing pages.
*/
public function execute()
{
    switch ($_SERVER["REQUEST_METHOD"]) {
        case "POST":
            $this->handlePostRequest();
            break;
    }
}
/**
 * POST requests must be forwarded to the system and the responses must be passed to the client to ensure correct landing page integration.
 */
private function handlePostRequest()
{
    // first fetch the csrf-token
    $this->fetchCsrfToken();

    // read the POST data sent by the landing page
    $requestBody = @file_get_contents("php://input");
    $requestData = json_decode($requestBody);

    // optional: enhance the request data with the IP address for tracking purposes
    $requestData->IpAddress = $_SERVER["REMOTE_ADDR"];

    // optional: add the campaign id to connect all landing page interactions to your campaign
    // $requestData->CampaignId = "your-campaign-id";

    // send the prepared request data to the system
    $requestString = json_encode($requestData);
    $response = $this->sendHttpRequest("POST", $this::BASE_PATH . $this::RESULT_HEADERS_PATH, $requestString);
private function fetchCsrfToken()
{
    $this->sendHttpRequest("HEAD", $this::BASE_PATH, null);
    if (! $this->csrfToken) {
        // HEAD request failed -> fallback using GET
        $this->sendHttpRequest("GET", $this::BASE_PATH, null);
    }
}

private function sendHttpRequest($method, $path, $body)
{
    // first create stream context
    $context = $this->createStreamContext($method, $body);

    // perform http request
    $response = file_get_contents($path, false, $context);

    if ($response === false) {
        // request failed - print error for analysis
        $error = error_get_last();
        if (is_array($error)) {
            echo $error['message'];
        } else {
            echo $error;
        }
    }

    // process response headers
    $this->readResponseHeaders($http_response_header);

    // return response
    return $response;
}

private function createStreamContext($method, $body)
{
    // basic authorization uses base64 encoded credentials
    $credentials = base64_encode($this::CREDENTIALS);

    // configure stream context
    $context = stream_context_create(
        array('http' => array('method' => $method,
                                'content_type' => 'application/json',
                                'credentials' => $credentials, 'timeout' => 15)));

    // return stream context
    return $context;
}
// build http request headers
$headers = array(
    "Authorization: Basic " . $credentials,
    "Accept: application/json",
    "Content-Type: application/json"
);

if ($this->cookies) {
    // add remembered cookies
    array_push($headers, "Cookie: " . $this->cookies);
}

// add x-csrf-token header for fetching or using the already fetched token
$csrfToken = ($this->csrfToken ?: "Fetch");
array_push($headers, "x-csrf-token: " . $csrfToken);

// build complete options array
$options = array(
    "http" => array(
        "header" => $headers,
        "method" => $method,
        "content" => $body,
        "ignore_errors" => true,
        "max_redirs" => 0
    )
);

// return stream context using the built options
return stream_context_create($options);

/**
 * This method processes the HTTP response headers
 * in order to read the fetched CSRF-Token and cookies.
 * @param array $responseHeaders
 */
private function readResponseHeaders($responseHeaders)
{
    // loop response headers
    foreach ($responseHeaders as $responseHeader) {
        // split header name from value
        $parts = explode(" ", $responseHeader);

        // handle response header based on name
        switch (strtolower($parts[0])) {
            case "HTTP/1.0":
                // status code
                http_response_code($parts[1]);
                break;
            case "x-csrf-token:":
                // save fetched csrf-token
                $this->csrfToken = $parts[1];
                break;
            case "set-cookie:":
                // set received cookies
                $this->cookies .= $parts[1];
                break;
        }
    }
}

// initialize the integration class and start the processing
$landingPageIntegration = new LandingPageIntegration();
$landingPageIntegration->execute();
System User Authentication

The result OData service `CUAN_CONTENT_PAGE_RESULT_SRV` can only be called by users with the corresponding authorization. For this reason, your Web server implementation must include a user authentication containing a technical user with role `SAP_CEI_CONTENT_PAGE_RSLT` assigned.

The example PHP script above shows the authentication via an HTTP header named `Authorization` using `Basic` authentication (user/password).

Contact Identification

The landing page integration offers different ways to identify the Web user who visits the landing page. The following usage scenarios are supported:

- **Scenario A**: The Web user is anonymous (unknown)
  In this scenario, the user cannot be identified on the landing page.

- **Scenario B**: The Web user has accessed the landing page using a tracking link in an SAP Hybris Marketing email.
  Scenario B does not require any additional implementation effort. The landing page script performs the required actions autonomously. If the landing page is accessed using a SAP Hybris Marketing email, the link contains a tracking ID that is sent along with the data requests. This ID is used to identify the user that received the email.

Prefill Contact Data

When a Web user who accesses a landing page is identified, it is possible to prefill data for the `Input` and `Permission` elements on the landing page. Selecting the `Prefill Contact Data` checkbox allows the landing page elements to be filled with data for the identified contact, which is maintained in the SAP Hybris Marketing system.

To support the prefill of contact data, the Web server implementation needs to pass the response data from the SAP Hybris Marketing system to the landing page (web client) that initiated the request. The SAP Hybris Marketing system provides all necessary data for landing pages with the `Prefill Contact Data` setting. There is no additional effort for the implementation, apart from the forwarding of response data.

Optional Attributes

In order to complete the landing page integration, you can enhance the OData requests with the following optional attributes:

- **IPAddress**
  The IP address of the web client visiting a landing page can be saved in order to have additional evidence that the user submitted the landing page, and gave marketing permissions and contact data.

- **CampaignId**
The campaign ID can be supplied to connect the interactions created out of the landing page to a specific SAP Hybris Marketing campaign. If the landing page is opened with a URL parameter `sap-campaign-id` with its value set to the ID, it is automatically added to all landing page requests. This connection will also be created if the landing page is opened out of a SAP Hybris Marketing email sent as part of a campaign.

### 3.2 Outbound

**Interaction Center** [page 32]
**Extensions** [page 32]
**Personalized Commerce** [page 33]
**Mobile, Social, and Digital Channel** [page 41]

### 3.2.1 Interaction Center

**SAP Contact Center and External Call Centers** [page 32]
With this integration you can forward call lists and call activities to a connected system of SAP Contact Center or of any other external call center.

#### 3.2.1.1 SAP Contact Center and External Call Centers

With this integration you can forward call lists and call activities to a connected system of SAP Contact Center or of any other external call center.

For more information, see on the SAP Help Portal under [https://help.sap.com/mkt](https://help.sap.com/mkt) » **SAP Hybris Marketing On Premise** » **SAP Hybris Marketing Product Page** » **Installation and Upgrade** » **Installation Guide** » **Setting up Call Center Integration with SAP Contact Center**.

### 3.2.2 Extensions

**Executing Campaigns Externally** [page 33]
With this integration you can execute a campaign with its target group in an external system and get back campaign success.
3.2.2.1 Executing Campaigns Externally

With this integration you can execute a campaign with its target group in an external system and get back campaign success.


3.2.3 Personalized Commerce

Consuming Recommendation Models Using OData [page 33]

The PROD_RECO_RUNTIME_SRV OData service enables customer channels to receive recommendations generated by Recommendation.

3.2.3.1 Consuming Recommendation Models Using OData

The PROD_RECO_RUNTIME_SRV OData service enables customer channels to receive recommendations generated by Recommendation.

The PROD_RECO_RUNTIME_SRV OData service enables customer channels to receive recommendations generated by Recommendation.

Prerequisites

- You have assigned the Marketing - Recommendation Integration communication scenario to your communication user in Maintain Communication Users.
- To receive the recommendations, you must call the service using the deep insert functionality of OData. For more information about the deep insert functionality of OData, see http://www.help.sap.com. Choose Technology SAP Gateway. Choose a release and then Application Help. In SAP Library, choose SAP NetWeaver Gateway Developer Guide OData Channel Advanced Features Deep Insert.

Details of Service Entity

Root URL: https://<Server>:<Port>/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/RecommendatioScenarios

Request Mode: POST

Entity Data Model: RecommendationScenarios
The nested structure of the entities that can be navigated to from the `RecommendationScenarios` entity are as follows:

- `RecommendationScenarios`
  - `Scenarios`
    - `LeadingObjects`
    - `BasketObjects`
  - `ContextParams`
  - `ScenarioHashes`
  - `ResultObjects`

### RecommendationScenario Entity Parameters

The following table contains the parameters of the `RecommendationScenario` entity:

Table 6:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserId</td>
<td>The ID of the user who performs the interaction, for example, customer ID or contact ID.</td>
<td>Edm.String</td>
<td>50</td>
<td>TRUE</td>
</tr>
<tr>
<td>UserType</td>
<td>The type of user who performs the interaction, for example, SAP Hybris Commerce Consumer or SAP Hybris Marketing Interaction Contact.</td>
<td>Edm.String</td>
<td>20</td>
<td>TRUE</td>
</tr>
<tr>
<td>ExternalTracking</td>
<td>A flag that implies external tracking of impressions using the PostImpressions function import (Optional).</td>
<td>Edm.Boolean</td>
<td>1</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### Scenario Entity Parameters
The following table contains the parameters of the **Scenario** entity:

Table 7:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScenarioId</td>
<td>The scenario ID represents a model type and related usage information, for example, promotion model type and user type.</td>
<td>Edm.String</td>
<td>50</td>
<td>TRUE</td>
</tr>
<tr>
<td>HashId</td>
<td>A hash associated to a specific user. The hash accelerates retrieving recommendations from the cache of an optimized algorithm.</td>
<td>Edm.String</td>
<td>32</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

**Note**

You can use the **ProductRecoScenario** entity to enable your customer channel to choose **ScenarioId** from a value help. For more information, see *Enabling Value Help Entities* [page 40].

**LeadingObject Entity Parameters**

The following table contains the parameters of the **LeadingObject** entity:

Table 8:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeadingObjectId</td>
<td>The ID of the leading object, for example, material number.</td>
<td>Edm.String</td>
<td>50</td>
<td>TRUE</td>
</tr>
<tr>
<td>LeadingObjectType</td>
<td>A recommendation data source type that is defined to an ITEM data source class, for example, SAP Hybris Commerce Product.</td>
<td>Edm.String</td>
<td>30</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

**Note**

You can use the **ItemSourceTypes** entity to enable your customer channel to choose **LeadingObjectType** from a value help. For more information, see *Enabling Value Help Entities* [page 40].

**BasketObjectid Entity Parameters**
The following table contains the parameters of the `BasketObjectId` entity:

Table 9:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>BasketObjectId</td>
<td>The ID of the leading object, for example, material number.</td>
<td>Edm.String</td>
<td>50</td>
<td>TRUE</td>
</tr>
<tr>
<td>BasketObjectType</td>
<td>A recommendation data source type that is defined to an ITEM data source class, for example, SAP Hybris Commerce Product.</td>
<td>Edm.String</td>
<td>30</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

**i Note**

You can use the `ItemSourceTypes` entity to enable your customer channel to choose `BasketObjectType` from a value help. For more information, see Enabling Value Help Entities [page 40].

**ContextParam Entity Parameters**

The following table contains the parameters of the `ContextParam` entity:

Table 10:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContextId</td>
<td>The prefilter parameter ID.</td>
<td>Edm.Int32</td>
<td>n.a.</td>
<td>TRUE</td>
</tr>
<tr>
<td>ContextParamId</td>
<td>The parent prefilter parameter ID.</td>
<td>Edm.Int32</td>
<td>n.a.</td>
<td>FALSE</td>
</tr>
<tr>
<td>Value</td>
<td>The value of the prefilter parameter.</td>
<td>Edm.String</td>
<td>100</td>
<td>FALSE</td>
</tr>
<tr>
<td>ValueType</td>
<td>The value type of the prefilter parameter.</td>
<td>Edm.String</td>
<td>32</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

**ScenarioHashes Entity Parameters**

The following table contains the parameters of the `ScenarioHashes` entity:

Table 11:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScenarioId</td>
<td>The recommendation scenario ID.</td>
<td>Edm.String</td>
<td>50</td>
<td>TRUE</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Edm Core Type</td>
<td>Max Length</td>
<td>Key</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>------------</td>
<td>-----</td>
</tr>
<tr>
<td>HashID</td>
<td>A hash returned by the system that is associated to a specific user. The hash accelerates retrieving recommendations from the cache of an optimized algorithm.</td>
<td>Edm.String</td>
<td>32</td>
<td>TRUE</td>
</tr>
<tr>
<td>ExpiresOn</td>
<td>Expiry date of HashID.</td>
<td>Edm.DateTime</td>
<td>FALSE</td>
<td></td>
</tr>
<tr>
<td>ResultScope</td>
<td>The scope of the result. For example, Generic, Restricted, or Personalized.</td>
<td>Edm.String</td>
<td>1</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

**ResultObject Entity Parameters**

The following table contains the parameters of the `ResultObject` entity:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScenarioId</td>
<td>The recommendation scenario ID.</td>
<td>Edm.String</td>
<td>50</td>
<td>TRUE</td>
</tr>
<tr>
<td>ResultObjectType</td>
<td>A recommendation data source type that is defined to an ITEM data source class. For example, SAP Hybris Commerce Product.</td>
<td>Edm.String</td>
<td>30</td>
<td>TRUE</td>
</tr>
<tr>
<td>ResultObjectId</td>
<td>The ID of the result object, for example, material number.</td>
<td>Edm.String</td>
<td>50</td>
<td>TRUE</td>
</tr>
<tr>
<td>ResultObjectScore</td>
<td>The score of the result object.</td>
<td>Edm.Decimal</td>
<td>10.5</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

**PostImpressions Function Import Parameters**
The following table contains the parameters of the `PostImpressions` function import:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScenarioId</td>
<td>The recommendation scenario ID.</td>
<td>Edm.String</td>
<td>50</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>The timestamp of the impression.</td>
<td>Edm.DateTimeOffset</td>
<td>30</td>
</tr>
<tr>
<td>ImpressionCount</td>
<td>The total number of impression performed.</td>
<td>Edm.Int16</td>
<td></td>
</tr>
<tr>
<td>ItemCount</td>
<td>The total number of Item recommended.</td>
<td>Edm.Int16</td>
<td></td>
</tr>
</tbody>
</table>

If the `ExternalTracking` parameter in the `RecommendationScenario` entity is set to `TRUE`, as it is in the HTTP post request example below, SAP Hybris Marketing Cloud will not count the impressions for the recommendation scenario that is being solicited. To keep the number of impressions in SAP Hybris Marketing Cloud accurate, it is necessary for the external system to convey the impression count. To do so, an additional separate call must be made to increase the impression count. For example, if the scenario `INT_TEST` returns 3 items that were consumed once; the additional call would contain the following:

```
https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/
PostImpressions?ScenarioId='INT_TEST'&TimeStamp=datet imeoffset'2016-12-03T12:45:29Z'&ImpressionCount=1&ItemCount=3&saml2=disabled
```

**Example**

**HTTP Post Request Using Deep Insert Functionality of OData in JSON Encoding:**

```json
{
"UserId" : "40F2E9306E391ED59BDE581AFE71F329 ",
"UserType" : "COOKIE_ID",
"ExternalTracking" : true,
"Scenarios" : [
{
"ScenarioId" : "INT_TEST",
"HashId" : "D33DD1F71615D50334FB2F1043365430",
"LeadingObjects" : [
{
"LeadingObjectType" : "SAP_ERP_MATNR",
"LeadingObjectId" : "M-01"
}
],
"BasketObjects" : [
{
"BasketObjectType" : "SAP_ERP_MATNR",
"BasketObjectId" : "100-100"
}
]
}],
"ContextParams" : [],
"ScenarioHashes" : [],
"ResultObjects" : []
}
```

---

Integration Guide 1702
Integration
HTTP Post Response Payload in JSON Encoding:

```json
{
  "d": {
    "__metadata": {
      "id": "https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/RecommendationScenarios(UserId='',UserType='')",
      "uri": "https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/RecommendationScenarios(UserId='',UserType='')",
      "type": "PROD_RECO_RUNTIME_SRV.RecommendationScenario"
    },
    "UserId": "",
    "UserType": "",
    "ExternalTracking": true,
    "ScenarioHashes": [],
    "results": []
  },
  "__metadata": {
    "id": "https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ScenarioHashes('SAP_TOP_SELLERS_EMAIL_CAMPAIGN')",
    "uri": "https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ScenarioHashes('SAP_TOP_SELLERS_EMAIL_CAMPAIGN')",
    "type": "PROD_RECO_RUNTIME_SRV.ScenarioHash"
  },
  "ScenarioId": "SAP_TOP_SELLERS_EMAIL_CAMPAIGN",
  "HashId": "D33DD1F71615D50334FB2F1043365429",
  "ExpiresOn": "/Date(1478180969524)/",
  "ResultScope": "G"
},
"ResultObjects": [
  {
    "__metadata": {
      "id": "https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectId='100-100')",
      "uri": "https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectId='100-100')",
      "type": "PROD_RECO_RUNTIME_SRV.ResultObject"
    },
    "ScenarioId": "INT_TEST",
    "ResultObjectType": "SAP_ERP_MATNR",
    "ResultObjectId": "100-100",
    "ResultObjectScore": "1.00000"
  },
  {
    "__metadata": {
      "id": "https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectId='P-102')",
      "uri": "https://[sap-hybris-marketing-server]/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ResultObject(ScenarioId='INT_TEST',ResultObjectType='SAP_ERP_MATNR',ResultObjectId='P-102')",
      "type": "PROD_RECO_RUNTIME_SRV.ResultObject"
    },
    "ScenarioId": "INT_TEST",
    "ResultObjectType": "SAP_ERP_MATNR",
    "ResultObjectId": "P-102",
    "ResultObjectScore": "1.00000"
  }
]
```
### 3.2.3.1.1 Enabling Value Help Entities

Entities that enable you to choose recommendation scenario and item source type parameters from a value help.

The **PROD_RECO_RUNTIME_SRV** OData service enables customer channels to receive recommendations generated by **Recommendation**. The **RecommendationRecoScenarios** and **ItemSourceTypes** entities enable customer channels to choose **ScenarioID**, **LeadingObjectType**, or **BasketObjectType** parameters from a value help.

#### ProductRecoScenarios Entity

**Root URL:** https://<Server>:<Port>/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/

**Request Mode:** GET

**Premission:** PFCG role SAP_COM_CSR_0019

**ProductRecoScenario Entity Parameters**
The following table contains the parameters of the `ProductRecoScenario` entity:

**Table 14:**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScenarioId</td>
<td>The ID of the scenario.</td>
<td>Edm.String</td>
<td>50</td>
<td>TRUE</td>
</tr>
<tr>
<td>ScenarioDescription</td>
<td>The description of the scenario.</td>
<td>Edm.String</td>
<td>255</td>
<td>FALSE</td>
</tr>
<tr>
<td>Language</td>
<td>The language of the scenario description.</td>
<td>Edm.String</td>
<td>30</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

**ItemSourceTypes Entity**

**Root URL:** https://<Server>:<Port>/sap/opu/odata/sap/PROD_RECO_RUNTIME_SRV/ItemSourceTypes

**Request Mode:** GET

**Premission:** PFCG role `SAP_COM_CSR_0019`

**ItemSourceTypes Entity Parameters**

The following table contains the parameters of the `ItemSourceTypes` entity:

**Table 15:**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ItemSourceId</td>
<td>The ID of the item source.</td>
<td>Edm.String</td>
<td>2</td>
<td>TRUE</td>
</tr>
<tr>
<td>ItemSourceTypeDescription</td>
<td>The description of the item source type.</td>
<td>Edm.String</td>
<td>255</td>
<td>FALSE</td>
</tr>
<tr>
<td>ItemSourceObjectType</td>
<td>The object type of the item source.</td>
<td>Edm.String</td>
<td>30</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

**3.2.4 Mobile, Social, and Digital Channel**

**Google AdWords, Yahoo, and Bing [page 42]**

With this integration you can push paid search campaigns to Google AdWords.

**China Localization with Baidu Maps and WeChat [page 42]**

With this integration, you can segment WeChat followers by geographical location on Baidu Maps and you can connect your system with WeChat official accounts and get data such as profiles of WeChat followers
and interactions. In addition, you can create campaigns in SAP Hybris Marketing and execute them on WeChat.

### 3.2.4.1 Google AdWords, Yahoo, and Bing

With this integration, you can push paid search campaigns to Google AdWords.

**Google AdWords, Yahoo, and Bing**

For more information, see the SAP Help Portal under [https://help.sap.com/mkt](https://help.sap.com/mkt) and choose:

- Integration > Integration with Paid Search and HANA Cloud Integration
- Application Help > SAP Hybris Marketing > SAP Hybris Marketing Applications > Campaigns > Paid Search Integration

### 3.2.4.2 China Localization with Baidu Maps and WeChat

With this integration, you can segment WeChat followers by geographical location on Baidu Maps and you can connect your system with WeChat official accounts and get data such as profiles of WeChat followers and interactions. In addition, you can create campaigns in SAP Hybris Marketing and execute them on WeChat.

In detail you can integrate the following solutions with your system:

- Baidu Maps
- WeChat Followers and Interactions
- Analysis of WeChat Campaigns Carried Out via Shake Nearby
- WeChat Campaigns

For more information about the offered integrations, see the SAP Help Portal under [http://help.sap.com/mkt](http://help.sap.com/mkt)

- China Localization: Under Installation and Upgrade choose Installation and Configuration Guide > Configuring China Localization
- Baidu Maps: Under Application Help choose SAP Hybris Marketing > SAP Hybris Marketing Applications > Segmentation > Segmentation > Segmentation of WeChat Followers via Baidu Maps
- WeChat Followers and Interactions: Under Application Help choose SAP Hybris Marketing > SAP Hybris Marketing Applications > Contacts and Profiles > WeChat Followers and Interactions
The following WeChat-specific campaign categories are available after you have configured the china localization:

Table 16:

<table>
<thead>
<tr>
<th>Campaign Categories</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>WeChat - Automated Campaign (Once) (WCA)</td>
<td>WeChat Broadcast Message (WC_BROADCAST_MESSAG)</td>
</tr>
<tr>
<td></td>
<td>WeChat Direct Message (WC_DIRECT_MESSAGE)</td>
</tr>
<tr>
<td>WeChat - Automated Campaign (Periodic) (WCP)</td>
<td>WeChat Broadcast Message (WC_BROADCAST_MESSAG)</td>
</tr>
<tr>
<td></td>
<td>WeChat Direct Message (WC_DIRECT_MESSAGE)</td>
</tr>
<tr>
<td>WeChat - Trigger-Based Campaign (WCT)</td>
<td>WeChat Direct Message (WC_DIRECT_MESSAGE)</td>
</tr>
</tbody>
</table>

- Analysis of WeChat Campaigns Carried Out via Shake Nearby
  Under Application Help choose SAP Hybris Marketing > SAP Hybris Marketing Applications > Contacts and Profiles > Analysis of WeChat Campaigns Carried Out via Shake Nearby
- WeChat Campaigns
  Under Application Help choose SAP Hybris Marketing > SAP Hybris Marketing Applications > Campaigns > Creating WeChat Campaigns

3.3 Supportive Integrations

Service Provider Integrations for Emails and Text Messages [page 44]
  With you can connect your system with service providers such as SAP Mobile Services, to send emails and text messages.

SAP Consumer Insight 365 [page 44]
  With this integration you can derive filter criteria from SAP Consumer Insight 365 for using it in the segmentation.

SAP HANA Rule Framework (HRF) [page 44]
  With this integration you can get rule-based and heuristic scores such as simple scores and lead scores.

SAP Hybris customer journey manager add-on [page 45]
  With this integration you can define and orchestrate campaigns using segmentation profiles from SAP Hybris customer journey manager add-on.

SAP Hybris Digital Asset Management by OpenText [page 45]
  With this integration you can enrich your email campaign content with product pictures from SAP Hybris Digital Asset Management by OpenText.

SAP Hybris Product Content Management (PCM) [page 46]
  With this integration you can enrich email campaign content with product pictures from SAP Hybris Product Content Management.

SAP Visual Business and here.com [page 48]
With these integrations you can translate addresses to geo-coordinates and reverse, as well as, doing geospatial analysis for segmentation using the connected maps.

**Data Export and Data Import [page 50]**
An additional kind of integration is the import and export of data using CSV files.

**Upload and Download of Campaign Content for Editing in External HTML-Editors [page 56]**
With this integration you can upload and download your campaign content to edit it in external HTML editors such as Adobe Dreamweaver, Photoshop, and Indesign. You can use this integration also for translation and test purposes.

### 3.3.1 Service Provider Integrations for Emails and Text Messages

With you can connect your system with service providers such as SAP Mobile Services, to send emails and text messages.

The following service provider are currently available:
- SAP Mobile Services (Emails and Text Messages)
- Amazon (Emails)
- Sybase (Emails)


### 3.3.2 SAP Consumer Insight 365

With this integration you can derive filter criteria from SAP Consumer Insight 365 for using it in the segmentation.


### 3.3.3 SAP HANA Rule Framework (HRF)

With this integration you can get rule-based and heuristic scores such as simple scores and lead scores.


- ![Installation Guide](http://help.sap.com/mkt) and look for Using Rule-Based Tasks
- ![Upgrade Guide](http://help.sap.com/mkt) and look for Using Rule-Based Tasks (1508)
Specific delivery unit of SAP HANA, therefore no special integration is required.

### 3.3.4 SAP Hybris customer journey manager add-on

With this integration you can define and orchestrate campaigns using segmentation profiles from SAP Hybris customer journey manager add-on.

For more information, see on the SAP Help Portal under [SAP Hybris Marketing On Premise](https://help.sap.com/mkt) > SAP Hybris Marketing Product Page > Integration > Integration for SAP Hybris Customer Journey Manager.

### 3.3.5 SAP Hybris Digital Asset Management by OpenText

With this integration you can enrich your email campaign content with product pictures from SAP Hybris Digital Asset Management by OpenText.

For more information, see [Setting up the Integration of Digital Asset Management Systems](#) [page 45].

#### 3.3.5.1 Setting up the Integration of Digital Asset Management Systems

Digital asset management (DAM) systems provide catalogs of digital images, videos, documents, music etc. Digital assets can be searched easily by keywords. With the integration of DAM systems you can directly access images for use in static image links in an email. SAP delivers standard settings for the following DAM systems:

- Hybris Product Content Management
- SAP Hybris Digital Asset Management by OpenText

Typically, the systems are activated and the credentials are set during the execution of the technical configuration. You can call up transaction ME_DAM to change the activation status and set the credentials afterwards.

If a system was activated with transaction ME_DAM, its RFC destination has to be set up manually.

If a system was activated during technical configuration, its empty RFC destination is created either with name CEI_ME_DAM_HYBRIS or with name CEI_ME_DAM_OPENTEXT, depending on the DAM system activated. Afterwards, the RFC destination has to be configured manually.

To perform this manual step, start transaction [Configuration of RFC Connections](#) (SM59) and choose HTTP Connections to External Server.
RFC Destination CEI_ME_DAM_HYBRIS for Hybris Product Content Management

Go to Technical Settings and maintain the following:

- **Target Host**: host name of the Hybris PCM system
- **Service No.**: port number for the API calls (i.e. 9001 for http and 9002 for https)
- **Path Prefix**: `/rest/v1/<site>/products/` with `<site>` being the BaseSite.uid property of the Web site (see documentation for Hybris Product Content Management).

Go to Logon & Security and maintain the following:

- **Logon Procedure**: Choose Do Not Use a User or Basic Authentication with User and Password depending on the configuration of the DAM system
- **Security Options**: Choose SSL Inactive or SSL Active with SSL Certificate depending on the configuration of the DAM system

Go to Special Options.

- **HTTP Cookies**: Set Accept Cookies to Yes (All)

RFC Destination CEI_ME_DAM_OPENTEXT for SAP Hybris Digital Asset Management by OpenText

Go to Technical Settings and maintain the following:

- **Target Host**: Host name of the OpenText system
- **Service No.**: Port number for the API calls (i.e. 80 for http and 443 for https)
- **Path Prefix**: none

Go to Logon & Security and maintain the following:

- **Logon Procedure**: Choose Do Not Use a User
- **Security Options**: Choose SSL Inactive or SSL Active with SSL Certificate depending on the configuration of the DAM system

Go to Special Options.

- **HTTP Cookies**: Set Accept Cookies to Yes (All).

3.3.6 SAP Hybris Product Content Management (PCM)

With this integration you can enrich email campaign content with product pictures from SAP Hybris Product Content Management.

For more information, see Setting up the Integration of Digital Asset Management Systems [page 47].
3.3.6.1 Setting up the Integration of Digital Asset Management Systems

Digital asset management (DAM) systems provide catalogs of digital images, videos, documents, music etc. Digital assets can be searched easily by keywords. With the integration of DAM systems you can directly access images for use in static image links in an email. SAP delivers standard settings for the following DAM systems:

- Hybris Product Content Management
- SAP Hybris Digital Asset Management by OpenText

Typically, the systems are activated and the credentials are set during the execution of the technical configuration. You can call up transaction ME_DAM to change the activation status and set the credentials afterwards.

If a system was activated with transaction ME_DAM, its RFC destination has to be set up manually.

If a system was activated during technical configuration, its empty RFC destination is created either with name CEI_ME_DAM_HYBRIS or with name CEI_ME_DAM_OPENTEXT, depending on the DAM system activated. Afterwards, the RFC destination has to be configured manually.

To perform this manual step, start transaction Configuration of RFC Connections (SM59) and choose HTTP Connections to External Server.

RFC Destination CEI_ME_DAM_HYBRIS for Hybris Product Content Management

Go to Technical Settings and maintain the following:

- **Target Host**: host name of the Hybris PCM system
- **Service No.**: port number for the API calls (i.e. 9001 for http and 9002 for https)
- **Path Prefix**: /rest/v1/<site>/products/ with <site> being the BaseSite.uid property of the Web site (see documentation for Hybris Product Content Management).

Go to Logon & Security and maintain the following:

- **Logon Procedure**: Choose Do Not Use a User or Basic Authentication with User and Password depending on the configuration of the DAM system
- **Security Options**: Choose SSL Inactive or SSL Active with SSL Certificate depending on the configuration of the DAM system

Go to Special Options.

- **HTTP Cookies**: Set Accept Cookies to Yes (All)
RFC Destination CEI_ME_DAM_OPENTEXT for SAP Hybris Digital Asset Management by OpenText

Go to Technical Settings and maintain the following:

- **Target Host**: Host name of the OpenText system
- **Service No.**: Port number for the API calls (i.e. 80 for http and 443 for https)
- **Path Prefix**: none

Go to Logon & Security and maintain the following:

- **Logon Procedure**: Choose Do Not Use a User
- **Security Options**: Choose SSL Inactive or SSL Active with SSL Certificate depending on the configuration of the DAM system

Go to Special Options.

- **HTTP Cookies**: Set Accept Cookies to Yes (All).

### 3.3.7 SAP Visual Business and here.com

With these integrations you can translate addresses to geo-coordinates and reverse, as well as, doing geospatial analysis for segmentation using the connected maps.

For more information, see SAP Help Portal [http://help.sap.com/mkt](http://help.sap.com/mkt) \>
SAP Hybris Marketing On Premise \>
SAP Hybris Marketing Product Page \>
Installation and Upgrade \>
Installation Guide \>
and look for Geospatial Segmentation.

**Note**

SAP only provides the interfaces and configuration options which allow you to connect the map visualization and geocoding services. The usage of here.com is not part of your end-user license agreement with SAP. It is your responsibility to check and/or adapt the default configuration.

### 3.3.7.1 Setting up the Geospatial Segmentation and Map Preview

This function allows you to view the distribution of geolocations on a map, for example, the distribution of customers in a region. You can then create new segments based on areas defined in the map.
Prerequisites:

Geocoordinates

To use geospatial segmentation, you need data in the form of geographic coordinates stored in your system. Addresses, for example, must first be converted to coordinates and then uploaded to the system. You can segment on different kinds of coordinates, such as the coordinates of interactions or shops in a certain area.

Access to Map Provider


To use the geospatial segmentation option with the map visualization, your browser requires access to the internet domain “here.com”. The domain provides the map data that is used in the geospatial segmentation option. Check, whether the users in question have access to this domain and consider the implications of communicating with servers outside your firewall.

Enabling a Geolocation Attribute

If you use a pair of numeric attributes to define the longitude and the latitude, set the Semantic Type of the longitude attribute in Segmentation Configuration, Segmentation Objects and Attributes Assigned Data Sources Geolocation Attribute Semantics.

Use one of the following options:

- To process geolocations with coordinate (0,0) as actual geo points, set the longitude attribute to Geo Point.
- To omit geolocations with coordinate (0,0) as initial values, set the longitude attribute to Geo Point without (0,0).

To complete the setting, define the latitude attribute as dependent on the longitude attribute. To do this, assign the latitude attribute under Semantics Name of Dependent Attribute to the longitude attribute.

Note

For attributes of type ST_POINT or ST_GEOMETRY, the semantic type is automatically set.

Preview Type Assignment

By default, the Geospatial Map preview (SAP.CH_GEOSPATIAL_MAP) is assigned to the Geolocation attribute of type GEO_POINT. Note that you can only assign attributes of this type to the Geospatial Map preview, and vice versa.
Using a Custom Attribute

If you want to use the geospatial segmentation based on custom business data, make sure that your attribute universe provides the required geolocation information.

Use one of the following two options to model the according SAP HANA database tables, and SAP HANA information models:

- Define a column of (SQL data) type `ST_POINT` or `ST_GEOMETRY` in your SAP HANA database table. Expose the column to your SAP HANA information model.
- In your SAP HANA database table, provide the longitude and latitude in two different numerical columns as a geolocation tuple. Expose the columns to your SAP HANA information model.

In addition to the modeling, configure the semantic type of the geolocation attribute in `Segmentation Configuration`.

3.3.8 Data Export and Data Import

An additional kind of integration is the import and export of data using CSV files.

Master Data

A central information concerning the import of master data is the Data Management Upload Interfaces guide. It contains information about:

- Contacts
- Interactions
- Companies
- Product Categories and Products
- Marketing Locations
- Marketing Beacons

For more information on how to import master data an external system, see the guide for data management upload interfaces on the SAP Help Portal under https://help.sap.com/mkt/SAP Hybris Marketing On Premise SAP Hybris Marketing Product Page Integration Data Management Upload Interfaces.

Other Data To Upload or Download

• Exporting Ad Campaigns and Target Groups (Segments)
  With this integration you can export ad campaign and target groups (segments) to an external system such as a demand-side or data management platform (DSP or DMP).
  For more information on how to import master data an external system, see the guide for data management upload interfaces on the SAP Help Portal under https://help.sap.com/mkt > SAP Hybris Marketing On Premise > SAP Hybris Marketing Product Page > Integration > Data Management Upload Interfaces.

• Exporting Target Groups and Target Group Member Data [page 51]

• Importing Brands and Custom Dimensions
  With this integration you can import brand and custom planning dimension data in your system.
  For more information, see Importing Brands Using CSV Upload [page 54] and Importing Custom Dimensions Using CSV Upload [page 54].

• Importing Business Documents for Contacts and Interactions
  With this integration you can import business documents and business documents for contacts and interactions from an external system.
  For more information, see:
  ○ SAP Service Marketplace http://service.sap.com/rds-ymkt
  ○ SAP Service Marketplace http://service.sap.com/rds-dlh

• Importing Data for Marketing Executive Dashboard
  With this integration you can import data for your marketing executive dashboard.

• Importing Offers
  For more information, see Import of Offers Using an OData Service [page 70].

  Exporting Target Groups and Target Group Member Data [page 51]
  With this integration you can export target groups to an external system.

  Create Export File [page 52]
  This action is for creating an export file. This enables you to export the campaign member data to execute the campaign using an external agency, for example.

  Importing Brands Using CSV Upload [page 54]
  Importing Custom Dimensions Using CSV Upload [page 54]

3.3.8.1 Exporting Target Groups and Target Group Member Data

With this integration you can export target groups to an external system.

For more information on how to export target group data, see:
  • Exporting Target Groups [page 52]
  • Create Export File [page 52]
3.3.8.1.1 Exporting Target Groups

Within SAP Hybris Marketing you have the possibility to export the data of your target group members as follows:

1. Create your export definitions in the system under **Campaigns > Export Definitions**.
2. Then you can create a campaign with a campaign category that allows to create export files, and create your export file on the **Automation** facet.
3. You can also create an export file for target group member data by choosing the export definition in the **Export** facet of the **Target Group** subworkset.


- **Campaigns > Campaigns**
- **Campaigns > Export Definitions**
- **Segmentation > Target Groups > Target Groups Details**

3.3.8.2 Create Export File

This action is for creating an export file. This enables you to export the campaign member data to execute the campaign using an external agency, for example.

You can use the **Export** action on all levels of a campaign automation for the campaign categories **Automated Campaign (Once)**, **Automated Campaign (Periodic)**, and **Trigger-Based Campaign**.

**Prerequisites**

- You have created an export definition in the **Export Definitions** app with the same segmentation object as used for your target group.
- To have the action available in trigger-based campaigns and on the follow-up triggers of target-group-based campaigns the following class must be assigned to the corresponding action in Customizing for SAP Hybris Marketing under **Campaigns > Campaign > Define Campaign Categories and Actions**:

<table>
<thead>
<tr>
<th>Action ID</th>
<th>Action Name</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPORT</td>
<td>Create Export File</td>
<td>CL_CUAN_MKT_EXEC_EXPORT</td>
</tr>
</tbody>
</table>
Parameters

- **Export Definition**: Select the export definition to get the required fields in the export file.
- **File Name Part**: This field is used to specify a part of the file name that is added to the automatically created name: `<System><Client>CPG-<Id of campaign>-<Number of trigger>-<Number of Action>-<File Name Part>-<Timestamp>` and helps you to find the files on your server again.
- **New File After**: This parameter is only visible if the action is used in a follow-up trigger or start trigger of trigger-based campaign. With this parameter, a new file will be created every given days (24 hours).
- **Delay Export**: This parameter is only visible if the action is used on a follow-up trigger or start trigger of trigger-based campaign. With this parameter it is possible to delay the reaction on triggers, such as *Email Opened* for a certain time. Otherwise, the action is performed immediately.
- **Delay Export by**: This parameter is only visible if the action is used on a follow-up trigger or start trigger of trigger-based campaigns, and the parameter *Delay Export* is set. The delay defines that the entry for a trigger is written into the export file after the given days, assuming 24 hours per day.

**Example**

Trigger happens on March, 25th, 10:00 a.m. and **Delay Export by** is set to 1 Day. So the entry is written in the export file on March, 26th, 10:00 a.m.

- **Export at**: This parameter is only visible if the action is used on a follow-up trigger or start trigger of trigger-based campaign and parameter *Delay Export* is set. The field defines the point of time, when the export file entry for the trigger should be written on the delayed day.

**Example**

1. Trigger A happens on March, 25th, 10:00 a.m., trigger B happens on March, 25th, 11:00 a.m., **Delay Export by** is set to 1 Day, and **Export at** is set to 5:00 p.m. So both triggers are written into export file on March, 26th, 5:00 p.m.
2. Trigger A happens on March, 25th, 10:00 a.m., trigger B happens on March, 25th, 11:00 a.m., **Delay Export by** is set to 0 Days (left empty), and **Export at** is set to 7:30 p.m. So both triggers are written into export file on March, 25th, 7:30 p.m.
3. Trigger A happens on March, 25th, 10:00 a.m., trigger B happens on March, 25th, 11:00 a.m., **Delay Export by** is set to 0 Days (left empty), and **Export at** is set to 8:45 a.m. So both triggers are written into export file on March, 26th, 8:45 a.m.

**Monitoring Export Files**

After the campaign has been executed you can track and open the file exports using *Download File*.

For periodic campaigns note that if you want to access a history of your exports in addition to the latest export file, you can select the **Export History** setting in Customizing for SAP Hybris Marketing under *Campaigns > Export Definitions > Define Global Settings for Data Export*. This allows the system to generate details for historic export files.
3.3.8.3 Importing Brands Using CSV Upload

For uploading brands, the user must be assigned the following authorization objects as defined in composite role SAP_MARKETING_BUS_ADMIN_USER:

Table 18:

<table>
<thead>
<tr>
<th>Authorization Object</th>
<th>Object Name (HPA_OBJ)</th>
<th>Action Name (HPA_ACTION)</th>
<th>Activity (ACTVT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPA_OBJ</td>
<td>HPA_BRAND</td>
<td>--</td>
<td>02 (Change)</td>
</tr>
<tr>
<td>HPA_ACTION</td>
<td>HPA_BRAND</td>
<td>IMPORT_BRANDS</td>
<td>16 (Execution)</td>
</tr>
</tbody>
</table>

The CSV file for uploading brand data must comply with the following guidelines:

- Enter the data for your upload after the header row.
- A header row that contains attribute names is mandatory.
- Ensure that no unknown attribute names are included in the header row, for example, due to typos. Unknown attribute names and the corresponding data are treated as comments and are ignored in the upload.
- As row separators, you can use CLRF for Microsoft Windows systems and CR for Unix systems.
- Ensure that there are no empty rows above your data. Delete empty rows or start with an asterisk (*) so that they are ignored in the upload.
- Comment rows are allowed only above the header row. They must start with an asterisk (*)
- As column separators, you can use a comma (,), semicolon (;), or tab.
- You must not enter any data in columns without an attribute name in the header row.
- Save the document as a CSV file.

The following is an example of a CSV file for uploading brand data:

```
Sample Code

ID,LANGUAGE,DESCRIPTION
BRAND1,E,Brand 1
BRAND2,E,Brand 2
BRAND3,E,Brand 3
```

ID, LANGUAGE, DESCRIPTION is the header row.

3.3.8.4 Importing Custom Dimensions Using CSV Upload

Custom dimensions can be used in budget planning. Before uploading custom dimensions, you must define the custom dimensions in Customizing for SAP Hybris Marketing under Planning Budget Planning Define Custom Dimensions. You can define a maximum of 10 custom dimensions. The technical object name for each dimension is CUAN_CUSTOM_DIMENSION (SAP technical ID of custom dimension from 01 to 10). The mapping from the dimension ID and the technical ID is done automatically. For example, you have defined a custom dimension with the ID MyDimension. It will be mapped to the technical ID CUAN_CUSTOM_DIMENSION_01.
For uploading custom dimensions, the user must be assigned the following authorization objects as defined in composite role SAP_MARKETING_BUS_ADMIN_USER:

Table 19:

<table>
<thead>
<tr>
<th>Authorization Object</th>
<th>Object Name (HPA_OBJ)</th>
<th>Action Name (HPA_ACTION)</th>
<th>Activity (ACTVT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPA_OBJ</td>
<td>CUAN_CUSTOM_DIMENSION</td>
<td>--</td>
<td>02 (Change)</td>
</tr>
<tr>
<td></td>
<td>(SAP technical ID of custom dimension from 01 to 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPA_ACTION</td>
<td>CUAN_CUSTOM_DIMENSION</td>
<td>IMPORT_CUSTOM_DIMENSION</td>
<td>16 (Execution)</td>
</tr>
<tr>
<td></td>
<td>(SAP technical ID of custom dimension from 01 to 10)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The CSV file for uploading custom dimension data must comply with the following guidelines:

- Enter the data for your upload after the header row. A header row that contains attribute names is mandatory.
- Ensure that no unknown attribute names are included in the header row (for example, due to typos). Unknown attribute names and the corresponding data are treated as comments and are ignored in the upload.
- As row separators, you can use \texttt{CLRF} for Microsoft Windows systems and \texttt{CR} for Unix systems.
- Ensure that there are no empty rows above your data. Delete empty rows or start with an asterisk (*) so that they are ignored in the upload.
- Comment rows are allowed only above the header row. They must start with an asterisk (*).
- As column separators, you can use a comma (,), semicolon (;), or tab.
- You must not enter any data in columns without an attribute name in the header row.
- Ensure that the data corresponds to the respective settings you made in Customizing for SAP Hybris Marketing under \texttt{Planning $\rightarrow$ Budget Planning $\rightarrow$ Define Custom Dimensions}.
- Ensure that the last comment of this file is the ID of the custom dimension for which you want to upload values.
- Save the document as a CSV file.
- Replace the last comment of the file with the ID of a custom dimension you defined in Customizing for SAP Hybris Marketing under \texttt{Planning $\rightarrow$ Budget Planning $\rightarrow$ Define Custom Dimensions}, whose values are going to be uploaded from this file.

The following is an example of a CSV file for uploading custom dimension data:

```
*LoB
ID;LANGUAGE;DESCRIPTION
01;E;Small
02;E;Medium
L03;E;Large
```

Where LoB is a custom dimension defined in Customizing for SAP Hybris Marketing under \texttt{Planning $\rightarrow$ Budget Planning $\rightarrow$ Define Custom Dimensions}.
3.3.9 Upload and Download of Campaign Content for Editing in External HTML-Editors

With this integration you can upload and download your campaign content to edit it in external HTML editors such as Adobe Dreamweaver, Photoshop, and Indesign. You can use this integration also for translation and test purposes.

For more information, see on the SAP Help Portal http://help.sap.com/mkt under [SAP Hybris Marketing On Premise] > [SAP Hybris Marketing Product Page] > [Application Help] > [SAP Hybris Marketing] > [SAP Hybris Marketing Applications] > [Campaigns] > [Content Studio] > [Templates, Marketing Emails and Text Messages for Campaigns] > [Create a Personalized Email or Email Template] > [Design a Personalized Email or Email Template]
4 APIs and OData Services

Extend your system using an API or OData service.

Channel Affinity [page 57]
A graph that displays the affinity score for various communication channel.

Import Business Partners Using Odata Service [page 58]

Generic OData Service for Offer Discovery [page 63]

Importing Data of Campaigns and Target Groups Using an OData Service [page 68]
With the OData Service CUAN_INITIATIVE_SRV you can retrieve certain attributes of campaigns and target groups, for example, for the recommendation scenario.

Import of Offers Using an OData Service [page 70]
Offer data can be maintained using the corresponding app in the system, but they can also be imported from other systems using a public OData application programming interface (API). You can use the public OData service CUAN_OFFER_IMPORT_SRV to upload (import) and read external offers. The upload of offer data is always started through the Import Headers entity and a deep insert on the Offer entity.

Import of Campaign Execution Plans Using an OData Service [page 82]

Import of Actual and Committed Spend Using an OData Service [page 85]

Edit Messages Used in Campaigns in External Systems Using a Public OData Service [page 91]
Download content of messages in an external system and upload the changed or new content.

4.1 Channel Affinity

A graph that displays the affinity score for various communication channel.

This is a graph that displays the preferred communication channel, such as email, Facebook and so on, for a selected target group. The score represents the response of the target group to the various communication channels. Depending on the result, a marketer can choose the preferred communication channel for engaging with the contact in future.

The scores for email is calculated based on the rules set in the Score Builder app. The scores for other channels of type Social Media is calculated based on the KPIs defined for each channel.

This feature is available in the Target Group Quick Launch. Select a target group and choose Channel Affinity to view the preferred channel. For more information, see .

Note
- This feature is available only for Static and Dynamic target groups.
- If a contact’s email address is not available or if the contact selected email opt-out, the contact’s score is 0, representing no email response.
4.2 Import Business Partners Using Odata Service

4.2.1 Overview

OData service CUAN_BUSINESS_PARTNER_IMPORT_SRV is used for standard SAP Hybris Marketing Cloud integration with SAP ERP. It is used in marketing-driven and sales-driven processes to replicate SAP ERP data to SAP Hybris Marketing Cloud interaction contacts.

For more information about the integration scenario, see in the SAP Hybris Marketing Cloud application help.

For more information about external interfaces that SAP Hybris Marketing Cloud provides for creating or updating interaction contacts, interactions, interests, corporate accounts, product categories, and products, see http://help.sap.com/mkt SAP Hybris Marketing On Premise ➤ SAP Hybris Marketing Product Page ➤ Integration ➤ Data Management Upload Interfaces.

OData service CUAN_BUSINESS_PARTNER_IMPORT_SRV can also be used to create SAP Hybris Marketing Cloud interaction contacts from any source system. In contrast to OData service CUAN_IMPORT, OData service CUAN_BUSINESS_PARTNER_IMPORT_SRV supports the change of interaction contacts. Each interaction contact is identified by the key of the business partner in the external system.

4.2.2 Technical Prerequisites

OData service CUAN_BUSINESS_PARTNER_IMPORT_SRV is available as part of the standard integration with SAP ERP. For more information, see


4.2.3 Basic Concepts

OData service CUAN_BUSINESS_PARTNER_IMPORT_SRV supports only batch processing and operation PATCH (MERGE) on the entity type InteractionContact. Other operations, such as create, update or delete are not supported.

Batch requests allow grouping multiple operations into a single HTTP request payload.
Batch requests are submitted as a single HTTP POST request to the $batch endpoint of a service as described in http://www.odata.org/documentation/odata-version-2-0/uri-conventions/. The batch request must contain a content-type header specifying a content type of `multipart/mixed` and a boundary specification.

A PATCH (MERGE) request updates only the properties indicated in the request body and leaves everything untouched that was not mentioned.

All properties that are not to be changed, can be omitted. The transmitted properties are merged with the data already stored in SAP Hybris Marketing Cloud.

If the OData service is not accessible – for example due to missing authorization, or because the system is not available – a corresponding HTTP status code is returned.

If the OData service is accepted by the gateway component in the SAP Hybris Marketing Cloud system, the HTTP status code 204 is returned. Potential processing errors are recorded in the SAP Hybris Marketing Cloud system in the Integration Errors app, where they can be monitored, restarted and discarded.

### 4.2.4 Structure of OData Service

**CUAN_BUSINESS_PARTNER_IMPORT_SRV**

The **CUAN_BUSINESS_PARTNER_IMPORT_SRV** OData service consists of the following entity sets and entity types:

<table>
<thead>
<tr>
<th>Entity Set</th>
<th>Entity Type</th>
<th>Entity Type Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InteractionContacts</td>
<td>InteractionContact</td>
<td>Interaction contacts refer to contacts in SAP Hybris Marketing Cloud. A contact is a natural person who interacts with a company. Contact data is collected and merged from several sources into the master data tables within SAP Hybris Marketing Cloud. We distinguish between contacts, corporate contacts, consumers and suspects to define the business relationship of a contact to a company.</td>
</tr>
</tbody>
</table>

The OData service **CUAN_BUSINESS_PARTNER_IMPORT_SRV** supports OData batch processing. Interaction contact data can be transferred by the OData PATCH (MERGE) operation.
Request Header

Table 21:

<table>
<thead>
<tr>
<th>Header Field</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sap-Cuan-SequenceId</td>
<td>Unique technical identifier of the imported data.</td>
<td>Edm.String</td>
<td>30</td>
<td>X</td>
</tr>
<tr>
<td>Sap-Cuan-RequestTimestamp</td>
<td>Timestamp of the data</td>
<td>Edm.DateTime</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>Sap-Cuan-SequenceNumber</td>
<td>Sequence number of the request. This number is normally incremented each time a new request for the same sequence id is created.</td>
<td>Edm.Int16</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>Sap-Cuan-SourceSystemType</td>
<td>Type of the source system</td>
<td>Edm.String</td>
<td>20</td>
<td>X</td>
</tr>
<tr>
<td>Sap-Cuan-SourceSystemId</td>
<td>Identifier of the source system</td>
<td>Edm.String</td>
<td>23</td>
<td>X</td>
</tr>
</tbody>
</table>

The header fields Sap-Cuan-SequenceId and Sap-Cuan-RequestTimestamp or Sap-Cuan-SequenceNumber are used to check the sequence of the received data. Data with a timestamp older or sequence number lower than data already imported, is ignored.

The Sap-Cuan-SourceSystemType and Sap-Cuan-SourceSystemId fields allow you to distinguish between different source systems.

*Either Sap-Cuan-RequestTimestamp or Sap-Cuan-SequenceNumber must be provided together with Sap-Cuan-SequenceId.

InteractionContact

Only the properties Id and IdOrigin are mandatory. All properties that are not to be changed can be omitted.

Table 22:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Mandatory</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>ID of Interaction Contact</td>
<td>Edm.String</td>
<td>255</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Edm Core Type</td>
<td>Max Length</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------</td>
<td>---------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----</td>
</tr>
<tr>
<td>IdOrigin</td>
<td>Origin of Interaction Contact</td>
<td>Edm.String</td>
<td>20</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Name</td>
<td>Full Name</td>
<td>Edm.String</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FirstName</td>
<td>First Name</td>
<td>Edm.String</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastName</td>
<td>Last Name</td>
<td>Edm.String</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TitleCode</td>
<td>Title Code</td>
<td>Edm.String</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RegionCode</td>
<td>Region Code</td>
<td>Edm.String</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>City</td>
<td>Edm.String</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PostalCode</td>
<td>Postal Code</td>
<td>Edm.String</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street</td>
<td>Street</td>
<td>Edm.String</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HouseNumber</td>
<td>House Number</td>
<td>Edm.String</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LanguageCode</td>
<td>Language Code</td>
<td>Edm.String</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GenderCode</td>
<td>Gender Code</td>
<td>Edm.String</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaritalStatusCode</td>
<td>Marital Status Code</td>
<td>Edm.String</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IndustryCode</td>
<td>Industry Code</td>
<td>Edm.String</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DepartmentCode</td>
<td>Department Code</td>
<td>Edm.String</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FunctionCode</td>
<td>Function Code</td>
<td>Edm.String</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EmailAddress</td>
<td>E-Mail Address</td>
<td>Edm.String</td>
<td>241</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>Phone Number</td>
<td>Edm.String</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MobilePhoneNumber</td>
<td>Mobile Phone Number</td>
<td>Edm.String</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FaxNumber</td>
<td>Fax Number</td>
<td>Edm.String</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>Date of Birth</td>
<td>Edm.DateTime</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IsContact</td>
<td>Indicates whether person acts as contact for an account</td>
<td>Edm.Boolean</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Edm Core Type</td>
<td>Max Length</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------</td>
<td>------------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----</td>
</tr>
<tr>
<td>IsConsumer</td>
<td>Indicates whether person is a consumer</td>
<td>Edm.Boolean</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obsolete</td>
<td>Obsolete</td>
<td>Edm.Boolean</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WebUri</td>
<td>Web Site</td>
<td>Edm.String</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latitude</td>
<td>Latitude</td>
<td>Edm.Decimal</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Longitude</td>
<td>Longitude</td>
<td>Edm.Decimal</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IcType</td>
<td>Interaction Contact Type: 01 person 02 company</td>
<td>Edm.String</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompanyId</td>
<td>ID of Company</td>
<td>Edm.String</td>
<td>255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompanyIdOrigin</td>
<td>Origin of Company</td>
<td>Edm.String</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MatchId</td>
<td>ID of matched Entity</td>
<td>Edm.String</td>
<td>255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MatchIdOrigin</td>
<td>Origin of matched Entity</td>
<td>Edm.String</td>
<td>20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The property `IdOrigin` indicates the source of the `Id`. It is maintained in the Self-Service Configuration app `Define Origins of Contact ID`. To replicate SAP ERP data to SAP Hybris Marketing Cloud, interaction contacts origin `SAP_ERP_CONTACT` or `SAP_ERP_CUSTOMER` is used.

`CompanyId` and `CompanyIdOrigin` are used to create a relationship between a contact and a company. `MatchId` and `MatchIdOrigin` are used to associate an interaction contact to data already created with different Id and IdOrigin in SAP Hybris Marketing Cloud, for example data transmitted from a different source system.

An interaction contact can be classified as contact (property: `IsContact = true`) for persons acting as contact for an account (B2B), and as a consumer (property: `IsConsumer = true`) that acts as an account (B2C). A person can be both a contact and a consumer at the same time.

### 4.2.5 Field Extensibility

In addition to the pre-delivered attributes, you can add customer-specific fields using the Custom Fields and Logic app. For more information about how to do this, see [http://help.sap.com/s4hana](http://help.sap.com/s4hana) > SAP S/4HANA Cloud <latest version> > Product Assistance > Generic Information > General Functions for the Key User > Extensibility > Custom Fields and Logic.
New fields can be added for the following BusinessContexts:

- Marketing: Interaction Contact
- Marketing: Person
- Marketing: Company

If the field is added to the BusinessContexts Person or Company, the respective IcType (01 for Person and 02 for Company) must be filled in the payload of new interaction contacts.

### 4.3 Generic OData Service for Offer Discovery

#### Overview

The public OData service CUAN_OFFER_DISCOVERY_SRV can be used to retrieve suitable offer content to consumers for a specific context. For example in a Web shop, such as SAP Hybris Commerce, the service can be used to determine offer content, such as a banner, to be displayed on a Web page in the shop. To find the most relevant offer content, a number of context parameters can be passed to the service. Possible contexts for a Web shop scenario include the following:

- Current user logged on to the Web shop to show personalized offers
- Browser language to determine offer content (such as images) in the correct language

**OData Version: 2.0**

**Root URL:** https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/

**Permissions:** PFCG role SAP_CEI_OFFER_PUBLIC_API

**Support of OData Features:** See the following chapters for implementation details and search behavior of the OData services.

#### Entity Data Model

The following entity types provide input help for the identically or similarly named properties for the complex type OfferContentSearch:

- CommunicationMedium
- ContentPosition
- ContentType
- CustomerOrigin
- Language
- MarketingLocation
- OfferRecommendationScenario

The relationship between these entity types and the corresponding properties in the complex type OfferContentSearch is reflected in the respective annotations in the metadata document for the service.
Value list annotation describing the relationship between (value help) entity type CommunicationMedium and the corresponding filterable property CommunicationMediumId for the complex type OfferContentSearch.

Sample Code

```xml
<Annotations xmlns="http://docs.oasis-open.org/odata/ns/edm"
  name="CUAN_OFFER_DISCOVERY_SRV.OfferContent/Search/CommunicationMediumId">
  <Annotation Term="com.sap.vocabularies.Common.v1.ValueList">
    <Record>
      <PropertyValue String="CommunicationMediums"
        Property="CollectionPath"/>
      <PropertyValue Property="Parameters">
        <Collection>
          <Record
            Type="com.sap.vocabularies.Common.v1.ValueListParameterInOut">
            <PropertyValue Property="LocalDataProperty"/>
            <PropertyValue String="CommunicationMediumId"
              Property="ValueListProperty"/>
          </Record>
          <Record
            Type="com.sap.vocabularies.Common.v1.ValueListParameterDisplayOnly">
            <PropertyValue String="CommunicationMediumName"
              Property="ValueListProperty"/>
          </Record>
        </Collection>
      </PropertyValue>
    </Record>
  </Annotation>
</Annotations>
```

The retrieval of OfferContent entities in the OData service CUAN_OFFER_DISCOVERY_SRV is done by requesting the entity set OfferContents using a GET operation, using a $filter operator with the available parameters for the complex type OfferContentSearch. A direct retrieval of individual OfferContent entities using the key fields is not implemented. The following request parameters can be used as filter properties in the $filter-clause:

- **CustomerId**: The user or customer or consumer ID of the user logged on to the Web shop. If this parameter is not used in $filter or used with an empty string, the parameter CustomerOriginId is ignored.
- **CustomerOriginId**: The origin ID for interaction contacts defined in Customizing. If this parameter is not used in $filter or used with an empty string and the CustomerId is also used, the parameter value is defaulted to the delivered value SAP_HYBRIS_CONSUMER internally.
- **CommunicationMediumId**: The ID of a communication medium. If a CommunicationMediumId is passed to the OData service as a filter, only offer contents for that communication medium are retrieved.
- **LanguageId**: The ISO language code of the offer content. In a Web shop, this might correspond to the user’s logon language. If no language is passed to the OData service, the result contains all available languages.
- **Position**: Position in the Web shop where offers are to be displayed, such as Top or Bottom. This information must have been entered for the offer content.
- **RecommendationScenarioId**: If you create rules in the Manage Offer Recommendation app for a recommendation scenario to find the best offers, use this field to assign the ID of the recommendation scenario to be used. If you do not make an entry in this field, the system uses the simple determination logic.
- **ContentMediumTypeId**: The ID of a content type, such as "01" for content type "Image". If this parameter is not used in $filter the value is defaulted to "01" internally. Only offer contents matching the specified content medium type ID are retrieved.

- **MarketingLocationKey**: One or more marketing location identifiers (UUIDs) to search for offers that either have no marketing location assigned or have one of the specified marketing locations assigned.

### Extensibility and Offer Content Types

The OData service **CUAN_OFFER_DISCOVERY_SRV** supports both "Offer Header Data" and "Offer Content Data" extensibility using the **Custom Fields and Logic** application. For more information, see . It is possible to read the values of extensibility fields when reading **OfferContent** entities. However, it is not possible to define filters for these extension fields.

For any content type defined in the app **Manage your Solution**, associated fields can be defined in the app **User Interface Adaptation**. The service provides the information about which fields belong to which offer content type in the following two ways:

- The value help entity type **ContentType** has a navigational property **AvailableFields** to read the list of available fields in the entity type **OfferContent**.
- Field-control fields are part of the entity type **OfferContent**. These fields control the visibility of the corresponding data field in the following way:
  - If the corresponding data field is part of the returned content type the value is 1 (meaning read-only)
  - If the corresponding data field is not part of the returned content type the value is 0 (meaning hidden)

### Search Behavior

The **CustomerId** and **CustomerOriginId** parameters can only accept a single value for the filter operation $filter with the operator EQ. Additional filters using these parameters are ignored. Other operators are ignored and set to EQ. If a range operator (for example BT) is used, the lower boundary value is used. The higher boundary value is ignored.

The parameter **CustomerId** is used for the eligibility check to match target group members in the system that are assigned to offers.

The search result contains 0-n **OfferContent** entities which are active on the date and time of the actual request and have only valid marketing location assignments. An offer is active at a particular point in time if the status is **Released** and if the point in time is found during one of the offer visibility (valid or visible) ranges.

A marketing location assigned to an offer is valid with respect to this offer if, and only if the following is true:

- The marketing location is not closed (deleted).
- The marketing area for both objects is equal.
- The offer end date, in UTC, is found during the validity period of the location.

The result set is sorted internally using a ranking that is based on the filterable properties **CustomerId** and **MarketingLocationKey**: Offers are assigned the highest ranking if the **CustomerId** is part of any assigned target group and the given locations are assigned to the offer. Conversely, offers with no target group and no location assignment are assigned the lowest ranking. These offers are valid for any customer in any location.

Additional sorting parameters can be passed to the service using $orderby, but are applied to the internal sorting after the logic described above.
Request

Calling the OData Service Operation

The OData API is only to be called using $batch, so that the query can be encrypted in the HTTP request body and to avoid URL overflows. A request to the OData service to retrieve up to ten OfferContent entities with filters on CustomerId, CommunicationMedium Language and Position could then be as follows:

Example HTTP request for offer content retrieval

Note

To improve readability, the following example HTTP requests and responses do not show all the details. Some metadata information is for example omitted in the JSON responses and URLs are shown without encoding. For example, spaces are not replaced by %20.

```
POST /sap/opu/odata/SAP/CUAN_OFFER_DISCOVERY_SRV/$batch
Request Headers:
accept          application/json
content-type    multipart/mixed;boundary=batch_1e29-6867-0e8e
Request Body:
--batch_1e29-6867-0e8e Content-Type: application/http Content-Transfer-Encoding: binary
GET OfferContents?$top=10&$filter=Search/CustomerId eq 'demo@hybris.com' and
Search/CommunicationMediumId eq 'ONLINE_SHOP' and Search/LanguageId eq 'FR' and
Search/Position eq 'TOP'
HTTP/1.1
Accept: application/json
Accept-Language: en-US
DataServiceVersion: 2.0
MaxDataServiceVersion: 2.0
--batch_1e29-6867-0e8e--
```

Response

Format: JSON

```
{
  "d": {
    "results": [],
    "Search": {
      "CustomerId": "",
      "CustomerOriginId": "",
      "CommunicationMediumId": "",
      "ContentMediumTypeId": "",
      "LanguageId": "",
      "Position": "",
      "RecommendationScenarioId": "",
      "MarketingLocationKey": ""
    },
    "OfferId": "0000004711",
  }
}
```
This response contains two OfferContent entities. The first entity was found because the customer (with id demo@hybris.com) was found in a target group that is assigned to offer 4711 and both the language, the communication medium and the position of the entity match the search query. The second OfferContent entity was found because the offer has no target group assigned and the language, the communication medium and the position match the corresponding filter values of the entity. The content of offer 4711 is ranked first according to the sorting rules described above.
4.4 Importing Data of Campaigns and Target Groups Using an OData Service

With the OData Service CUAN_INITIATIVE_SRV you can retrieve certain attributes of campaigns and target groups, for example, for the recommendation scenario.

Overview

With the OData Service CUAN_INITIATIVE_SRV you can retrieve certain attributes of campaigns and target groups, for example, for the recommendation scenario.

Details of the Service Entity

- **URLs:**
- **Request Mode:** GET
- **Permissions:** PFCG role Marketing - Recommendation Integration (SAP_COM_CSR_0019)
- **Entity Data Model:** CUAN Initiative (CUAN_INITIATIVE)

Permissions: PFCG role SAP_CEI_OFFER_PUBLIC_API

Support of OData Features: See the following chapters for implementation details and search behavior of the OData services.

Entity Type

Initiative Entity Type

Table 23:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the campaign</td>
<td>Edm.String</td>
<td>40</td>
<td>No</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the campaign</td>
<td>Edm.String</td>
<td>n.a.</td>
<td>No</td>
</tr>
<tr>
<td>InitiativeId</td>
<td>The identifier of the campaign</td>
<td>Edm.String</td>
<td>10</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Property Descriptions

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitiativeExt</td>
<td>The external identifier of the campaign</td>
<td>Edm.String</td>
<td>10</td>
<td>No</td>
</tr>
<tr>
<td>LifeCycleStatus-StatusCode</td>
<td>The life cycle status code of the campaign</td>
<td>Edm.String</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>LifeCycleStatus-StatusDescription</td>
<td>The life cycle status description of the campaign</td>
<td>Edm.String</td>
<td>60</td>
<td>No</td>
</tr>
<tr>
<td>Search-SearchTerm</td>
<td>The search term of the campaign</td>
<td>Edm.String</td>
<td>n.a.</td>
<td>No</td>
</tr>
<tr>
<td>Search-TileFilterCategory</td>
<td>The tile filter category of the campaign</td>
<td>Edm.String</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>Category-CategoryCode</td>
<td>The category code of the campaign</td>
<td>Edm.String</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>Category-CategoryDescription</td>
<td>The category description of the campaign</td>
<td>Edm.String</td>
<td>60</td>
<td>No</td>
</tr>
<tr>
<td>Category-CategoryType</td>
<td>The category type of the campaign</td>
<td>Edm.Int16</td>
<td>n.a.</td>
<td>No</td>
</tr>
<tr>
<td>Filter-InteractionContactId</td>
<td>The interaction contact identifier of the campaign</td>
<td>Edm.String</td>
<td>n.a.</td>
<td>No</td>
</tr>
<tr>
<td>Filter-InteractionContactIdOrigin</td>
<td>The interaction contact identifier origin of the campaign</td>
<td>Edm.String</td>
<td>20</td>
<td>No</td>
</tr>
</tbody>
</table>

### Target Group Entity Type

#### Table 24:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetGroupId</td>
<td>The identifier of the target group</td>
<td>Edm.String</td>
<td>10</td>
<td>Yes</td>
</tr>
<tr>
<td>CustomerMemberCount</td>
<td>The customer member count of the target group</td>
<td>Edm.Int32</td>
<td>n.a.</td>
<td>No</td>
</tr>
</tbody>
</table>
OData Service Call (GET) Examples Used in the Recommendation Scenario

Searching campaigns with a search term and additional filters:

```
Sample Code
$expand=TargetGroup&
$select=Name,Description,InitiativeId,InitiativeIdExt,LifecycleStatus,TargetGroup/
CustomerMemberCount&$filter=Search/SearchTerm eq 'tes' and Category/CategoryCode
eq '' and (Search/TileFilterCategory eq '1' or Search/TileFilterCategory eq '2')
```

Selecting a specific campaign to create a customer segment:

```
Sample Code
Initiatives('0000009108')/?$expand=TargetGroup&
$select=Name,Description,InitiativeId,InitiativeIdExt,LifecycleStatus,TargetGroup/
CustomerMemberCount
```

Getting campaigns for the current user with the specified filters:

```
Sample Code
$select=Name,Description,InitiativeId,InitiativeIdExt&$filter=Category/
CategoryCode eq '' and Search/TileFilterCategory eq '1' and (Filter/
InteractionContactId eq '1d998c85cc3d5205' or Filter/InteractionContactId eq
'john.dempsey@hana.com') and (Filter/InteractionContactIdOrigin eq 'EMAIL' or
Filter/InteractionContactIdOrigin eq 'COOKIE_ID')
```

4.5 Import of Offers Using an OData Service

Offer data can be maintained using the corresponding app in the system, but they can also be imported from other systems using a public OData application programming interface (API). You can use the public OData service CUAN_OFFER_IMPORT_SRV to upload (import) and read external offers. The upload of offer data is always started through the Import Headers entity and a deep insert on the Offer entity.

Entity Data Model


Table 25:

| OData Version: | 2.0 |

"CUSTOMER"
Table 26: Entity Data Model: ImportHeader

<table>
<thead>
<tr>
<th>Name</th>
<th>Is Key</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Creatable</th>
<th>Updatable</th>
<th>Sortable</th>
<th>Nullable</th>
<th>Filterable</th>
<th>Complex Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>TRUE</td>
<td>Edm.String</td>
<td>--</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>TimeStampe</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>--</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>UserName</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>12</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>SourceSystemType</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>3</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>SourceSystemId</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>20</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>ProcessAllOrNothing</td>
<td>FALSE</td>
<td>Edm.Boolean</td>
<td>--</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

Table 27: Entity Data Model: Offer

<table>
<thead>
<tr>
<th>Name</th>
<th>Is Key</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Creatable</th>
<th>Updatable</th>
<th>Sortable</th>
<th>Nullable</th>
<th>Filterable</th>
<th>Complex Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>OfferIdExt</td>
<td>TRUE</td>
<td>Edm.String</td>
<td>60</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>OfferIdOrigin</td>
<td>TRUE</td>
<td>Edm.String</td>
<td>30</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>Name</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>40</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>Description</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>--</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
</tbody>
</table>
If you have enabled extension fields for the import service using the app *Custom Fields And Logic*, these extension attributes are added to the offer entity for the import service accordingly.

### Table 28: Entity Data Model: Marketing Area

<table>
<thead>
<tr>
<th>Name</th>
<th>Is Key</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Creatable</th>
<th>Updatable</th>
<th>Sortable</th>
<th>Nullable</th>
<th>Filterable</th>
<th>Complex Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarketingAreaDescription</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>120</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>MarketingAreaID</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>40</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>ExternalStatus</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>2</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>ExternalStatusDescrip</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>60</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>StartDate</td>
<td>FALSE</td>
<td>Edm.DateTime</td>
<td>--</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>EndDate</td>
<td>FALSE</td>
<td>Edm.DateTime</td>
<td>--</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

### Table 29: Entity Data Model: External Offer Status

<table>
<thead>
<tr>
<th>Name</th>
<th>Is Key</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Creatable</th>
<th>Updatable</th>
<th>Sortable</th>
<th>Nullable</th>
<th>Filterable</th>
<th>Complex Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExternalStatus</td>
<td>TRUE</td>
<td>Edm.String</td>
<td>2</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
</tbody>
</table>
The OData service provides the basic CRUD services as follows:

The upload of offer data is always started through the Import Headers entity and, in order to provide bulk processing, a deep insert on the Offer entity (CREATE_DEEP_ENTITY).

It is also possible to read offers (OFFERS_GET_ENTITY), update offers (OFFERS_UPDATE_ENTITY) and delete offers (OFFERS_DELETE_ENTITY).

The fields of the OData entities have the following meaning:

- ImportHeader
  - Id: A technical ID of one import service execution. In case no value is provided by the caller, an ID is generated by system.
  - Timestamp: Timestamp of the import run. In case no value is provided by the caller, a timestamp is generated by the system.
  - UserName: Name of the user who started the import. In case no value is provided by the caller, the system uses system name.
  - SourceSystemType: The type of the source system (can be freely defined, could be, for example, CRM or ERP).
  - SourceSystemId: The ID of the source system. Can be freely defined.
  - ProcessAllOrNothing: This flag defines if, in case an error occurs, all imported offers are discarded or only the faulty ones. Default is true.

- Offer
  - OfferIdExt: The unique offer ID provided by the external system which serves, in the import scenario, as part of the internal offer key.
  - OfferIdOrigin: A unique identifier of the origin of the external offer. This origin ID also serves in the import scenario as part of the internal offer key. It should logically match the SourceSystemId and SourceSystemType of the ImportHeader entity type.
  - Name: Name of the external offer (freetext).
  - Description: Description of the external offer (freetext).
  - MarketingAreaDescription: The description of the marketing area which has to be known by the system. Based on this description, the import system determines the ID of the marketing area. Note that either the marketing area description or the marketing area ID has to be provided.
  - MarketingAreaID: The ID of the marketing area which has to be known by the system. Note that either the marketing area description or the marketing area ID has to be provided.
  - StartDate: The validity start date of the offer.
  - EndDate: The validity end date of the offer.
  - ExternalStatus: Status of the offer that can be defined by the external system (optional). The allowed values can be retrieved by the ExternalOfferStatus entity.
ExternalStatusDescription: Status description of the offer that can be defined by the external system (optional). The allowed values can be retrieved by the ExternalOfferStatus entity.

ExternalOfferStatus
- ExternalStatus: Status of the offer that can be defined by the external system.
- ExternalStatusDescription: Status description of the offer that can be defined by the external system.

MarketingArea
- MarketingAreaID: The ID of the marketing area which has to be known by the system
- MarketingAreaDescription: The description of the marketing area which has to be known by the system.

Calling the OData Service Operation

Create Request

The upload of offer data is started as a post request through the Import Headers entity and a deep insert on the Offer entity (CREATE_DEEP_ENTITY). The following example shows the coding for creating a single offer.

URL (POST): /sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/$batch

Example Coding:

```
--batch_ab9a-082c-ba1e
content-type: multipart/mixed; boundary=changeset_3f1a-4263-a9e6
--changeset_3f1a-4263-a9e6
Content-Type: application/http
Content-Transfer-Encoding: binary
POST ImportHeaders HTTP/1.1
Content-Type: application/json
Accept: application/json
{"Id":"",
"Timestamp":null,
"UserName":"
"SourceType":"ERP",
"SourceSystemId":"ABC_server4711",
"ProcessAllOrNothing":true,
"Offers":[
{"OfferIdExt":"123",
"OfferIdOrigin":"ERP_ABC",
"Name":"Name for offer123",
"Description":"Description for offer123",
"MarketingAreaDescription":"GLOBAL",
"StartDate":"Date(1432634400000)",
"EndDate":"Date(1432634400000)",
"__metadata":{"type":"CUAN_OFFER_IMPORT_SRV.Offer"},
"__metadata":{"type":"CUAN_OFFER_IMPORT_SRV.ImportHeader"}]
--changeset_3f1a-4263-a9e6--
--batch_ab9a-082c-ba1e--
```

Response

Response Codes

Example for response in case of successful creation:

```
<?xml version="1.0"?>Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.
```
HTTP/1.0 202 Accepted

HTTP/1.0

202

Error! Hyperlink reference not valid.

Accepted

Error! Hyperlink reference not valid.

content-type

multipart/mixed; boundary=6DC2635F65A1D2A5E4ACABABA4AF2DD3F0

Error! Hyperlink reference not valid.

content-length

2313

Error! Hyperlink reference not valid.

dataserviceversion

2.0

Error! Hyperlink reference not valid.

remote_addr

Error! Hyperlink reference not valid.

uri_scheme_expanded

NONE

Error! Hyperlink reference not valid.

RequestID

40F2E930666C1E595CE796538562D1B4

Error! Hyperlink reference not valid.

--4859F7791E3AAAFAFP50CEE164B9E904CC0
Content-Type: multipart/mixed; boundary=4859F7791E3AAAFAFP50CEE164B9E904CC1
Content-Length: 1459

--4859F7791E3AAAFAFP50CEE164B9E904CC1
Content-Type: application/http
Content-Length: 1290
content-transfer-encoding: binary
HTTP/1.1 201 Created

content-length

1051
location: https://ldciana.wdf.sap.corp:44300/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/ImportHeaders('40F2E930666C1ED597896C07B8BE5513C')
dataserviceversion: 2.0


Example for response in case of error during creation:

--9A4E16C8A49C41AC3E5AE52CC6A89ACD0

Example for response in case of error during creation:
Read Request

The read request is a get with the corresponding key:

URL (GET): /sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/Offers(OfferIdExt='123',OfferIdOrigin='ERP_ABC')

Response Codes

Example for response in case of successful read:

```xml
<?xml version="1.0"?>
<NAME>~response_line</NAME>
<VALUE>HTTP/1.0 200 OK</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~server_protocol</NAME>
<VALUE>HTTP/1.0</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~status_code</NAME>
<VALUE>200</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~status_reason</NAME>
<VALUE>OK</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~set-cookie</NAME>
<VALUE>sap-usercontext=sap-language=E&sap-client=100; path=/</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~set-cookie</NAME>
<VALUE>SAP_SESSIONID_ANA_100=iMWRIDVltcR1jRG31XMKQtI3EG9zOxHlswBA8ukw2mw%3d; path=/; secure; HttpOnly</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~content-type</NAME>
<VALUE>application/atom+xml;type=entry; charset=utf-8</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~content-length</NAME>
<VALUE>1164</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~dataserviceversion</NAME>
<VALUE>2.0</VALUE>
</_-IWFND_-SUTILPROPERTY>
<NAME>~sap-metadata-last-modified</NAME>
<VALUE>Wed, 14 Oct 2015 14:35:29 GMT</VALUE>
</_-IWFND_-SUTILPROPERTY>
```
<?xml version="1.0" encoding="UTF-8"?>
<entry>
  <id>https://ldciana.wdf.sap.corp:44300/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/Offers(OfferIdExt='222',OfferIdOrigin='ERP_ABC')</id>
  <title type="text">Offers(OfferIdExt='222',OfferIdOrigin='ERP_ABC')</title>
  <link title="Offer" rel="self" href="Offers(OfferIdExt='222',OfferIdOrigin='ERP_ABC')"/>
  <d:OfferIdExt>222</d:OfferIdExt>
  <d:OfferIdOrigin>ERP_ABC</d:OfferIdOrigin>
  <d:Name>Name for offer123</d:Name>
  <d:Description>Description for offer123</d:Description>
  <d:MarketingAreaDescription>Global</d:MarketingAreaDescription>
  <d:StartDate>2015-05-26T10:00:00.0000000</d:StartDate>
  <d:EndDate>2015-05-26T10:00:00.0000000</d:EndDate>
</entry>

Example for response in case of failed read:

<?xml version="1.0"?>
<entry>
  <id>https://ldciana.wdf.sap.corp:44300/sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/Offers(OfferIdExt='222',OfferIdOrigin='ERP_ABC')</id>
  <title type="text">Offers(OfferIdExt='222',OfferIdOrigin='ERP_ABC')</title>
  <link title="Offer" rel="self" href="Offers(OfferIdExt='222',OfferIdOrigin='ERP_ABC')"/>
  <d:OfferIdExt>222</d:OfferIdExt>
  <d:OfferIdOrigin>ERP_ABC</d:OfferIdOrigin>
  <d:Name>Name for offer123</d:Name>
  <d:Description>Description for offer123</d:Description>
  <d:MarketingAreaDescription>Global</d:MarketingAreaDescription>
  <d:StartDate>2015-05-26T10:00:00.0000000</d:StartDate>
  <d:EndDate>2015-05-26T10:00:00.0000000</d:EndDate>
</entry>
Update Request

The update request is a put with the corresponding key:

URL (PUT): /sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/Offers(OfferIdExt='123',OfferIdOrigin='ERP_ABC')
Response Codes

Example for response in case of successful update:

```
<?xml version="1.0"?>
<NAME>~response_line</NAME>
<VALUE>HTTP/1.0 204 No Content</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>~server_protocol</NAME>
<VALUE>HTTP/1.0</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>~status_code</NAME>
<VALUE>204</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>~status_reason</NAME>
<VALUE>No Content</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>content-length</NAME>
<VALUE>0</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>dataserviceversion</NAME>
<VALUE>2.0</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>~remote_addr</NAME>
```

Example for response in case of failed update:

```
<?xml version="1.0"?>
<NAME>~response_line</NAME>
<VALUE>HTTP/1.0 400 Bad Request</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>~server_protocol</NAME>
<VALUE>HTTP/1.0</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>~status_code</NAME>
<VALUE>400</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>~status_reason</NAME>
<VALUE>Bad Request</VALUE>
</_-IWFND_-SUTIL_PROPERTY>
<NAME>~remote_addr</NAME>
```
Delete Request

The delete request is a delete with the corresponding key:

URL (DELETE): /sap/opu/odata/sap/CUAN_OFFER_IMPORT_SRV/Offers(OfferIdExt='123',OfferIdOrigin='ERP_ABC')

Response Codes

Example for response in case of successful delete:

```xml
<?xml version="1.0"?>
<response_line>HTTP/1.0 204 No Content</response_line>
```
Example for response in case of failed delete:

```
<?xml version="1.0" encoding="UTF-8"?>
<CUAN_ODATA_OFFER_MSG/020>
<message xml:lang="en">Offer import is completed.</message>
<component_id>CEC-MKT-CEI-OFM</component_id>
<service_namespace>/SAP/</service_namespace>
<service_id>CUAN_OFFER_IMPORT_SRV</service_id>
<service_version>0001</service_version>
</CUAN_ODATA_OFFER_MSG/020>
```
4.6 Import of Campaign Execution Plans Using an OData Service

Campaign execution plans can be imported from other systems using a public OData application programming interface (API). You can use the public CUAN_MPO_IMPORT_SRV OData service to upload (import) campaign execution plans. The upload of campaign execution plans is always started through the ImportHeaders entity and a deep insert on the ExecPlanItem entity.

Entity Data Model

The following tables list the details of the Campaign Execution Plan import service entities.

Table 30:

<table>
<thead>
<tr>
<th>OData Version:</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root URL:</td>
<td>https://&lt;Server&gt;:&lt;Port&gt;/sap/opu/odata/SAP/CUAN_MPO_IMPORT_SRV/</td>
</tr>
<tr>
<td>Permissions:</td>
<td>PFCG role: SAP_CBI_MPO_EXEC_PLAN_IMPORT</td>
</tr>
<tr>
<td>Support of OData Features:</td>
<td>See the following chapters for implementation details and search behavior of the OData services.</td>
</tr>
</tbody>
</table>
Table 31: Entity Data Model: ImportHeader

<table>
<thead>
<tr>
<th>Name</th>
<th>Is Key</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Creatable</th>
<th>Updatable</th>
<th>Sortable</th>
<th>Nullable</th>
<th>Filterable</th>
<th>Complex Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>TRUE</td>
<td>Edm.String</td>
<td>--</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>--</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>UserName</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>12</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>SourceSystemType</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>3</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>SourceSystemId</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>20</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>ProcessAllOrNothing</td>
<td>FALSE</td>
<td>Edm.Boolean</td>
<td>--</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

Table 32: Entity Data Model: ExecPlanItem

<table>
<thead>
<tr>
<th>Name</th>
<th>Is Key</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Creatable</th>
<th>Updatable</th>
<th>Sortable</th>
<th>Nullable</th>
<th>Filterable</th>
<th>Complex Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>TRUE</td>
<td>Edm.String</td>
<td>10</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>OptimizationScenarioId</td>
<td>TRUE</td>
<td>Edm.String</td>
<td>20</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>CampaignId</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>10</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>FALSE</td>
<td>Edm.DateTime</td>
<td>--</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
<tr>
<td>InteractionContactId</td>
<td>FALSE</td>
<td>Edm.String</td>
<td>255</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

The OData service provides the basic CRUD services as follows:

The upload of data is always started through the ImportHeaders entity and, in order to provide bulk processing, a deep insert on the ExecPlanItem entity (CREATE_DEEP_ENTITY).
The fields of the OData entities have the following meaning:

- **ImportHeader**
  - **Id**: A technical ID of one import service execution. In case no value is provided by the caller, an ID is generated by system.
  - **Timestamp**: Timestamp of the import run. In case no value is provided by the caller, a timestamp is generated by the system.
  - **UserName**: Name of the user who started the import. In case no value is provided by the caller, the system uses system name.
  - **SourceSystemType**: The type of the source system and can be freely defined, for example, **CRM** or **ERP**.
  - **SourceSystemId**: The ID of the source system and can be freely defined.
  - **ProcessAllOrNothing**: In case an error occurs, this flag defines if all imported offers are discarded or only the faulty ones. Default is **true**.

- **ExecPlanItem**
  - **Id**
  - **OptimizationScenarioId**
  - **CampaignId**
  - **Timestamp**
  - **InteractionContactId**

## Calling the OData Service Operation

### Create Request

The upload of campaign execution plans is started as a post request through the **ImportHeaders** entity and a deep insert on the **ExecPlanItem** entity (**CREATE_DEEP_ENTITY**). The following example shows the coding for creating a campaign execution plan.

**URL (POST):** `/sap/opu/odata/sap/CUAN_MPO_IMPORT_SRV/ImportHeaders`

```json
POST data:
{
  "Id" : "",
  "Timestamp" : "2016-07-01T08:10:12",
  "SourceSystemType" : "EXT",
  "SourceSystemId" : "JMeter_Auto",
  "ExecPlanItems" : [
    {
      "Id" : "",
      "OptimizationScenarioId" : "PHONE",
      "CampaignId" : "234",
      "Timestamp" : "2016-06-16T13:10:12",
      "InteractionContactId" : "3440B5B11ACE1EE693DCDDFB3B211B5"
    },
    {
      "Id" : "",
      "OptimizationScenarioId" : "PHONE",
      "CampaignId" : "321",
      "Timestamp" : "2016-07-16T17:11:03",
      "InteractionContactId" : "3440B5B11ACE1EE693E95CEE"
    }
  ]
}
```
Response

Example for response in case of successful creation:

```json
{
    "d": {
        "_metadata": {
            "id": "<system>:<port>/sap/opu/odata/sap/CUAN_MPO_IMPORT_SRV/ImportHeaders('E41D2DE534A01ED6A2F92AC2DD49165E')",
            "uri": "<system>:<port>/sap/opu/odata/sap/CUAN_MPO_IMPORT_SRV/ImportHeaders('E41D2DE534A01ED6A2F92AC2DD49165E')",
            "type": "CUAN_MPO_IMPORT_SRV.ImportHeader"
        },
        "Id": "E41D2DE534A01ED6A2F92AC2DD49165E",
        "Timestamp": "\Date(1467360612000)\",
        "SourceSystemType": "EXT",
        "SourceSystemId": "JMeter_Auto",
        "Username": "AUT_TESTER",
        "ProcessAllOrNothing": false,
        "ExecPlanItems": null
    }
}
```

4.7 Import of Actual and Committed Spend Using an OData Service

You upload actual and committed spend amounts from an external SAP ERP system into your system using OData service CUAN_ACTUAL_IMPORT_SRV.

Prerequisite

The SICF node for the external service name CUAN_ACTUAL_IMPORT_SRV has been maintained in the SAP Hybris Marketing gateway system (which is either the back-end system or a remote gateway system, depending on your setup). You can check in transaction SICF if the service is available (usually, the service is implemented during the standard installation process of SAP Hybris Marketing).
Authorization

Users who want to consume the CUAN_ACTUAL_IMPORT_SRV OData service, must have the following authorizations:

- **SICF node**
- **Start authorization for maintaining the respective business object; authorization object HPA_OBJECT:**
  - $\text{HPA\_OBJ} = \text{CUAN\_INITIATIVE}; \text{ACTVT = 03}$
  - $\text{HPA\_OBJ} = \text{CUAN\_MARKETING\_SPEND}; \text{ACTVT = 02}$
  - $\text{HPA\_ACTION = IMPORT\_ACTUAL}; \text{HPA\_OBJ = CUAN\_MARKETING\_SPEND}; \text{ACTVT = 16}$
- **Authorization to call the external service (role SAP\_CEI\_ACTUAL\_INTEGRATION in transaction PFCG) this role contains all required authorizations. You can copy and adapt this role according to your requirements.**

Mass Import

The OData protocol allows the import or update of one object record (one spend item) only. To achieve the mass create and mass update of records, a dummy entity (import header) is created by deep insert. You perform an insert on the entity ImportHeader and create actual spend items as subnodes of the import header. The metadata of the service is read by means of the OData call:

- **Request URI:** `/sap/opu/odata/sap/CUAN_ACTUAL_IMPORT_SRV/$metadata`
- **HTTP Method:** Get

Structure of OData Service CUAN_ACTUAL_IMPORT_SRV

The CUAN_IMPORT_SRV OData service consists of the following entity sets and entity types:

<table>
<thead>
<tr>
<th>Entity Set</th>
<th>Entity Type</th>
<th>Entity Type Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CampaignActuals</td>
<td>CampaignActual</td>
<td>Actual Spends</td>
</tr>
<tr>
<td>ImportHeaders</td>
<td>ImportHeaders</td>
<td>Technical Import Message Header</td>
</tr>
</tbody>
</table>

**Entity Type ImportHeader**

The entity type ImportHeader describes the technical header of an import of actual spends. The property Id is used as external reference number to identify the associated application log.
### Table 34: ImportHeader Meta Information

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Mandatory</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Used as external reference number in the application log</td>
<td>Edm.String</td>
<td>100</td>
<td>--</td>
<td>X</td>
</tr>
</tbody>
</table>

**Entity Type CampaignActual**

The entity type `CampaignActual` contains all attributes that are required to upload an actual spend records. If the values for a combination of source ID, campaign ID, spend type, spend item ID, and reference date are being uploaded for the first time, the corresponding values for this combination (amounts and currency) from the HTTP request are uploaded. If they are being uploaded a subsequent time, the corresponding values for this combination are updated with the values from the HTTP request.

Actual and committed spend can be created at any level. You are responsible to create data at the level that is relevant for the campaign and your business process. For example, you can create data at the following levels:

- Campaign level
- Campaign and spend level
- Campaign and spend item level

### Table 35: CampaignActual Meta Information

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Mandatory</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceId</td>
<td>The source ID indicates the source of the spend information, for example, it can indicate the agency from which the spend originates</td>
<td>Edm.String</td>
<td>30</td>
<td>--</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Mandatory</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>CampaignId</td>
<td>The campaign ID. Campaigns have been created in SAP Hybris Marketing, for example, in SAP Hybris Marketing Acquisition (Campaigns subworkset) or SAP Hybris Marketing Planning (Detailed Spend Planning subworkset).</td>
<td>Edm.String</td>
<td>10</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Edm Core Type</td>
<td>Max Length</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----</td>
</tr>
<tr>
<td>SpendType</td>
<td>Spend types have been defined in the Define Spend Type customizing activity</td>
<td>Edm.String</td>
<td>10</td>
<td>--</td>
<td>X</td>
</tr>
<tr>
<td>SpendItemId</td>
<td>The spend item ID. If specified, the spend item ID must exist in the campaign and the spend type must match. If no spend type is specified, the spend type will be derived from the spend item of the campaign</td>
<td>Edm.String</td>
<td>10</td>
<td>--</td>
<td>X</td>
</tr>
<tr>
<td>ReferenceDate</td>
<td>The date that is used for currency conversion. The reference date must be a valid date in the following format “YYYY-MM-DDT00:00:00”</td>
<td>Edm.DateTime</td>
<td>--</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Currency</td>
<td>Currency cost for the spend amount</td>
<td>Edm.String</td>
<td>5</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>ActualSpend</td>
<td>Actual costs that have been incurred from marketing activities</td>
<td>Edm.Decimal</td>
<td>15, 2</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>CommittedSpend</td>
<td>The amount of already known spend based on existing requests and orders for an item, for example, from a purchasing system</td>
<td>Edm.Decimal</td>
<td>15, 2</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Importing Actual and Committed Spend Using ODate Service

To upload actual spend the entity types ImportHeader.CampaignActuals is needed.

Additionally please consider the following for your upload:

- If the values for a combination of source ID, campaign ID, spend type, spend item ID, and reference date are being uploaded for the first time, the corresponding values for this combination (amounts and currency) from the local file are uploaded. If they are being uploaded a subsequent time, the corresponding values for this combination are updated with the values from the new local file.

- A period (.) must be used to separate decimals in amounts.

- If a field is optional and you do not want to include a value for it in the file, you still need to insert a comma (,) in place of the excluded value.

- Actual and committed spend can be uploaded at any level. You are responsible for uploading data at the level that is relevant for your campaign and your business processes. For example, you can upload data at the following levels:
  - Campaign level
  - Campaign and spend level
  - Campaign and spend item level

If there is invalid data in the local file, no actual and committed spend amounts are uploaded and saved.

If an error occurs, the complete file is rejected and no data is uploaded. Correct the content of the input file and execute the report again.

Example

- Request URI: /sap/opu/odata/sap/cuan_actual_import_srv/ImportHeaders
- HTTP Method: Post
- Example Request

```json
{
  "Id": "Example-01",
  "CampaignActuals": [
    {
      "SourceId": "",
      "CampaignId": "40144",
      "SpendType": "",
      "SpendItemId": "1",
      "ReferenceDate": "2015-11-30T00:00:00",
      "Currency": "USD",
      "ActualSpend": "50000.00",
      "CommittedSpend": "30000.00"
    },
    {
      "SourceId": "",
      "CampaignId": "40144",
      "SpendType": "",
      "SpendItemId": "2",
      "ReferenceDate": "2015-11-30T00:00:00",
      "Currency": "USD",
      "ActualSpend": "10000.00",
      "CommittedSpend": "10000.00"
    }
  ]
}
```
Success Message

After a successful upload of the actual spend, the status of the HTTP response is 201 Created and the system provides you with a success message in the following appearance:

```xml
    <id>https://wdciwe1.wdf.sap.corp:11100/sap/opu/odata/sap/cuan_actual_import_srv/ImportHeaders('Example-01')</id>
    <title type="text">ImportHeaders('Example-01')</title>
    <updated>2015-12-01T13:44:21Z</updated>
    <link href="ImportHeaders('Example-01')" rel="self" title="ImportHeader"/>
    <link href="ImportHeaders('Example-01')/CampaignActuals" rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/CampaignActuals" type="application/atom+xml;type=feed" title="CampaignActuals"/>
    <m:inline/>
</entry>
```

Error Handling

If the request fail due to some errors, the complete HTTP request is rejected and errors must be corrected before uploading again. In case of error the status of the HTTP response is 400 Bad Request and the system provides you with an error message in the following appearance.

```xml
<?xml version="1.0" encoding="utf-8"?><error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
    <code>SY/530</code>
    <message xml:lang="en">An exception was raised.</message>
    <innererror>
        <application>
            <component_id>CEC-MKT-CEI-MSM</component_id>
            <service_namespace>/SAP</service_namespace>
            <service_id>cuan_actual_import_srv</service_id>
            <service_version>0001</service_version>
        </application>
        <transactionid>565DF2DA9D1CD81AE1000000A445A06</transactionid>
        <timestamp>20151201135137.3145460</timestamp>
        <SAP_Transaction>Run transaction /IWFND/ERROR_LOG on SAP Gateway hub system and search for entries with the timestamp above for more details</SAP_Transaction>
        <SAP_Note>See SAP Note 1797736 for error analysis (https://service.sap.com/sap/support/notes/1797736)</SAP_Note>
    </innererror>
</error>
```
4.8 Edit Messages Used in Campaigns in External Systems Using a Public OData Service

Download content of messages in an external system and upload the changed or new content.

Overview

Messages used in campaigns, such as emails, email templates and text messages can be maintained using the app Messages and Email Templates, but can also be imported by other systems using this public OData application programming interface (API): CUAN_CUAN_MARKETING_MSG_IMP_EXP_SRV.

You can use the public OData service to download (export) the content blocks of a message, process (for example translate) the message in an external system and upload (import) the changed or new content blocks created to SAP Hybris Marketing.

OData Version: 2.0
Root URI: https://<Server>:<Port>/sap/opu/odata/SAP/CUAN_MARKETING_MSG_IMP_EXP_SRV/

Permissions: PFCG role SAP_CE1_MEM

For information about adding the OData service to a role, see the upgrade guide for SAP Hybris Marketing on the SAP Help Portal at: http://help.sap.com/mkt
SAP Hybris Marketing On Premise  
SAP Hybris Marketing Product Page  
Installation and Upgrade  
Upgrade Guide  
Adaptations of Roles and Authorization  
Adding an OData Service to a Role

Support of OData Features: See the following chapters for information about the implementation and behavior of the OData services.

Entity Data Model: Message

The `Message` entity represents the content of a message for a language.

The following table lists the details of the `Message` entity in the data model:

Table 36:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Unique message identifier</td>
<td>Edm.Int32</td>
<td>n.a.</td>
<td>Key, Searchable</td>
</tr>
<tr>
<td>Language</td>
<td>Language ISO code</td>
<td>Edm.String</td>
<td>2</td>
<td>Key, Searchable</td>
</tr>
<tr>
<td>LanguageName</td>
<td>Language name, such as “English”</td>
<td>Edm.String</td>
<td>16</td>
<td>None</td>
</tr>
<tr>
<td>Type</td>
<td>Type of message: “EM” (Email) “SMS” (Text Message) “TMP” (Email Template) “CON2 (Confirmation)</td>
<td>Edm.String</td>
<td>3</td>
<td>Searchable</td>
</tr>
<tr>
<td>TypeDescription</td>
<td>Type description, for example “Email Template”</td>
<td>Edm.String</td>
<td>60</td>
<td>None</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the message: “10” (Planned) “20” (Released) “60” (Outdated). Only messages with Status “10” can be changed.</td>
<td>Edm.String</td>
<td>2</td>
<td>Searchable</td>
</tr>
<tr>
<td>StatusDescription</td>
<td>Status description, such as “Planned”</td>
<td>Edm.String</td>
<td>60</td>
<td>None</td>
</tr>
</tbody>
</table>
Entity Data Model: **BlockContent**

The BlockContent property represents blocks that contain message content. BlockContent can be retrieved using the BlockContents navigation property in the Message entity.

There are two different use cases for the BlockContent entity: Block and Conditional Content:

**Block**

A block in the message together with its (default) content. Represented in the API by a BlockContent entity where the property ConditionName is initial.

The property Position defines the position of the block in the email, starting at 0.

**Conditional Content**

Block content that replaces the default content of the block when a condition is met. Represented in the API by a BlockContent entity with the property ConditionName set.

If more than one conditional content exists for a block, the property Position defines the sequence in which the corresponding conditions are evaluated, starting at 0.

The first conditional content in the sequence that has a condition that is evaluated as true is chosen to replace the default content. Subsequent conditional content is not considered.

The following table lists the details of the BlockContent entity in the data model:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Edm Core Type</th>
<th>Max Length</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageId</td>
<td>Unique message Identifier</td>
<td>Edm.Int32</td>
<td>n.a.</td>
<td>Key</td>
</tr>
<tr>
<td>Language</td>
<td>Language ISO code</td>
<td>Edm.String</td>
<td>2</td>
<td>Key, Creatable</td>
</tr>
<tr>
<td>BlockId</td>
<td>Unique Identifier of the block</td>
<td>Edm.String</td>
<td>20</td>
<td>Key, Creatable</td>
</tr>
</tbody>
</table>
Use Cases

This chapter describes how the OData API can be used to perform certain tasks.

Get List of Messages

A list of messages can be retrieved as a Message entity set. Filters for language, name, type and status can be applied.

Excluding “Html” from the list of selected properties will improve the response time.

Get Single Message

A single Message entity can be retrieved using its ID and language.

Get Block Contents of a Message

The BlockContent entity set associated with a Message entity can be retrieved using the BlockContents navigation property.

Create New Language

A new language is created implicitly by creating a BlockContent entity in this language.

Append Block at the End of the Message

Create BlockContent supplying properties:

- MessageId: ID of the message
- Language: ISO code of the language
- BlockId: New, must not yet exist within the message
- **ConditionName**: Left blank
- **Position**: Must be greater than the position of all other blocks in this language
- **Type**: SUBJECT, TEXT, PRODUCT, OFFER or PROD_RECO
- **Content**: Default HTML content of the block
- **Width**: Optional, width of the block in % or pixels
- **StyleClasses**: Style classes to be attached to the block
- **Editable**: True if a standard user can edit the block

**Insert Block Between Other Blocks**

Create **BlockContent** entity as above with:

- **Position**: Position of block in the list (starting at 0)

Move subsequent blocks down by increasing the **Position** property for the corresponding **BlockContent** entities.

**Update Block Content**

Update the property **Content** in the corresponding **BlockContent** entity.

**Update Block Attributes**

Update the properties **Editable**, **Width**, or **StyleClasses** in the corresponding **BlockContent** entity.

**Move Block**

Update the property **Position** in the **BlockContent** entity that corresponds to the block that has been moved.

Update the property **Position** in the **BlockContent** entities that correspond to all blocks between the previous and the new position of the block.

**Delete Block**

Delete the **BlockContent** entity.

This will also delete all corresponding conditional content.

**Insert Conditional Content**

Create the **BlockContent** entity with

- **BlockId** of the block (must exist)
- **ConditionName** containing the name of an existing or new condition
- **Position** defining the sequence of condition evaluation

Update the property **Position** in the **BlockContent** entities that correspond to other conditional content for this block, if necessary.

**Update Conditional Content**

Update the property **Content** in the corresponding **BlockContent** entity.

**Update Conditional Attributes**

Update the property **StyleClasses** in the corresponding **BlockContent** entity.

The properties **Width** and **Editable** are read-only for conditional content.
Change Condition Evaluation Order

Update the property Position in the BlockContent entities that correspond to affected conditional content.

Delete Conditional Content

Delete the corresponding BlockContent entity.

Delete Language

Delete all blocks for the language.

Constraints

The following constraints exist when changing block content:

- Only messages with status "10" (Planned) can be changed.
- Each language can only have one subject block.
- The block ID must be unique within a message across all languages.
- You cannot create conditional content unless default content is available.
- Each condition can only be used once in each block.
- The language must contain a valid ISO language code.
- The block type must be valid or empty.
- The HTML content must be valid and not contain scripts. For more information about valid HTML content, see in the chapter Prerequisites.
Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP’s gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP’s current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP’s gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).