PUBLIC
SAP Asset Manager
Document Version: 4.0 1911 – 2020-05-15

SAP Asset Manager Configuration Guide

THE BEST RUN SAP

Content

Document History

Before you begin reading this guide, be sure that you have the latest version. Find the latest version at https://help.sap.com/viewer/product/SAP_ASSET_MANAGER/p/en-US.

The following table provides an overview of the most important document changes.

Document Version	Date	Description of Changes
1.0	NOV 2019	Original release of the *SAP Asset Manager Configuration Guide*, version 1911
1.1	MAR 2020	Multiple minor changes, corrections, and additions Moved all push notification installation from the *SAP Asset Manager Installation Guide* to the *Push Framework Settings Procedures* section in this guide.
1.2	MAY 2020	Added note to first bullet point in the Supported Back-End Systems [page 6] topic.

1 SAP Asset Manager Overview

The *SAP Asset Manager Configuration Guide* is intended for system administrators, technical architects, implementation team members, and IT personnel involved in the installation, setup, and configuration of software for the application.

It is assumed that the personnel performing the installation and setup are familiar with SAP installation guidelines. SAP setup knowledge is helpful while carrying out the steps for the mobile setup of SAP.

Use the *SAP Asset Manager Configuration Guide* along with appropriate SAP documentation. Note that this guide only covers setting up and enabling the SAP Asset Manager mobile application.

1.1 Supported Back-End Systems

SAP Asset Manager 1911 supports the following back-end systems:

- SAP S/4HANA 1610 FPS 03 or higher

  > **i Note**
  >
  > For SAP S/4HANA on-premise 1909 systems, no ABAP add-on installation is required. Check 2493602, including the prerequisites section. For the SAP S/4HANA on-premise 1909 release, SAP Asset Manager 1911 is only available in SAP S/4HANA 1909 FPS01 and above releases.

- SAP Enhancement Package 7 for SAP ERP 6.0 Support Package 14 or higher
- SAP Enhancement Package 8 for SAP ERP 6.0 Support Package SP07 or higher

The same functions are available for both back-end systems.

For detailed information, see the following guides:

- Mobile Add-On for ERP Installation Guide
- Mobile Add-On for S/4HANA Installation Guide

1.2 SAP Asset Manager Functional Overview

SAP Asset Manager is a mobile solution for managing work orders, notifications, condition monitoring, and material consumption. The application also performs time management and failure analysis.

Regardless of connectivity, SAP Asset Manager allows remote employees to access, complete, and manage their assigned work orders and notifications through their mobile devices. With SAP Asset Manager, they have SAP back end data readily available including task lists, repair histories, reference documents, and geospatial data such as addresses and maps. Armed with more information, employees work smarter, have more work time, improve their first-time fix rates, and extend asset lives by conducting more preventative maintenance.

SAP Asset Manager comes packaged with a mobile add-on for SAP ERP and a mobile add-on for SAP S/4HANA. They offer tight integration and easier deployment without interference to or from your existing SAP system customizations or standard SAP objects. They provide you with full configuration, administration, and monitoring features that allow you to manage the SAP Asset Manager application from within your SAP system infrastructure.

The main features and functions available in SAP Asset Manager include the following.

Work Orders and Notifications

SAP Asset Manager supports the following standard SAP Plant Maintenance work order functionalities on the mobile device:

- Download work order assignments based on various assignment models
- Display work orders and their details, including repair histories
- Edit work orders and their details
- Create work orders and their details
- Complete work orders and update their status

The following standard SAP notification functionalities are supported on the mobile device:

- Download notification assignments based on various assignment models
- Display notifications and their details
- Edit notifications and their details
- Create notifications and their details
- Complete notifications and update their status

Maintenance Execution Data Capture

The following data related to maintenance execution can be captured from the mobile device:

- Maintenance order confirmation, goods issue for work orders
- Measurement readings for work orders
- Damage codes for notifications

Time Management

Maintenance technicians can use SAP Asset Manager to trace their time efficiently and accurately by entering the timesheet and the attendance records from the mobile device.

Single Sign-On (SSO)

Single Sign-On (SSO) allows the user to log into the SAP Asset Manager application from the client using SSO credentials without having to enter their back end user name and password. In addition, once logged in with SSO, you can access another mobile application without the need to log in again.

Documents

SAP Asset Manager supports viewing of master data or transaction data attachments on the mobile device. Documents include Microsoft Office files, PDFs, and other commonly used business documents, including videos, pictures, and audio files.

Downloading and uploading documents are supported for the following objects:

- Work orders
- Notifications
- Equipment
- Functional locations

The following standard SAP document storage options are supported:

- SAP Business Document Service (BDS)
- SAP Generic Object Service (GOS)
- SAP Document Management System (DMS)

Field Operations Worker Component

Inspection rounds with routing is an optional feature available as the SAP Asset Manager Field Operations Worker (FOW) component. FOW supports:

- Route and stop definitions
- Technical object assignments
- Measuring point assignments
- Measurement readings

Route and stop definition is implemented via the standard work order inspection round functionality.

Crew Management Component

SAP Asset Manager supports work crew management. This feature supports:

- Daily crew list definition
- Crew member and vehicle assignment
- Crew time management

Crew management is an optional feature available as the SAP Asset Manager Crew Management component.

Meter Management Component

SAP Asset Manager supports the industry solution for utilities meter management. The following standard features are supported:

- Meter installation (full or technical) via work order
- Meter replacement via work order
- Meter removal via work order
- Meter repair via work order

Customer Service Component

SAP Asset Manager supports customer service. This feature supports:

- Details of service engagements of the technician with the customer
- Details of business partners for the customer
- Technicians have access to contract and warranty information for the customer
- Mapping functionality for both customer addresses and partner address, provided the mobile device has internet access

Asset Central Component

Asset Central links production systems and assets with manufacturing and maintenance business processes to reduce operational and maintenance costs and increase assetuptime. Using Asset Central, you can use PdMS, or Predictive Maintenance and Service equipment indicators that allow you to identify the health status of your equipment.

1.2.1 Supported SAP Transaction Codes for SAP Asset Manager

SAP Asset Manager uses the SAP back end and specific SAP ERP transaction codes to help configure the application.

Transaction Codes	Definition
IW21	Create notifications

Transaction Codes	Definition
IW22	Edit notifications
IW23	Display notifications
IW31	Create work orders
IW32	Edit work orders
IW33	Display work orders
IW51	Create service notification
IW52	Change service notification
IW53	Display service notification
CAT2	Record time entries
IE03	Display equipment
IL03	Display functional location
MM03	Display material
IK13	Display measurement document
IK12	Change measurement document
IK11	Create measurement document
IW41	PM confirmation create
MIGO	Goods issue for work orders
EG30	IS utility meter management - device full replacement
EG31	IS utility meter management - device full installation
EG32	IS utility meter management - device full removal
EG33	IS utility meter management - device technical installation
EG36	IS utility meter management - device technical removal
EL28	IS utility meter management - meter reading result single entry

2 SAP Mobile Add-On for the SAP Configuration Panel

The SAP Mobile Add-On provides integration services for SAP Asset Manager. A central configuration tool known as the SAP Configuration Panel is provided to perform all configuration tasks related for the mobile application. The Configuration Panel is a browser-based application based on Web Dynpro ABAP.

2.1 Accessing the SAP Mobile Add-On for SAP Configuration Panel

Context

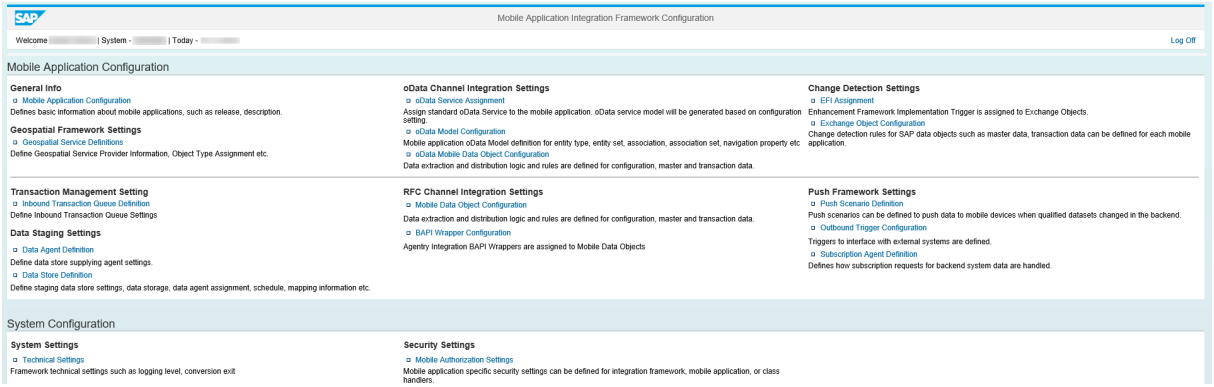
You can access the Configuration Panel either through SAP Customizing or using a transaction code directly. First, log into your back-end system, and then you can choose from the following two options:

Procedure

1. To access the ConfigPanel through *Customizing*, enter the transaction *spro* to open the *Customizing: Execute Project* screen. Select the *SAP Reference IMG* tab. Using the *SAP Customizing Implementation Guide* list, select *Agentry SAP Framework Configuration* > *System Settings* > *Define Mobile Applications* .
2. To access the ConfigPanel using a direct transaction code shortcut, enter */n/syclo/configpanel*.

Results

The *Mobile Application Integration Framework* home page is displayed in a browser.



2.2 Configuration Panel Overview

All configuration activities for the SAP Mobile Add-On are performed through the ConfigPanel.

Customization changes you make via the ConfigPanel can significantly impact the behavior of the SAP Mobile Add-On and the SAP Asset Manager application. Always follow SAP best practices, make changes and test them in the development and quality control systems before you transport the changes into your production landscape.

2.2.1 Standard Operations in the Configuration Panel

While configuration for each mobile application is unique, certain toolbar functions in the Configuration Panel are common and are available for all applications.

Mobile Application Filter

If more than one mobile application is available in the same system, you can use the filter function to only view items for a specific application. Find the filter option on any page where multiple applications are displayed.

To filter by application, click the arrow to the right of the *Defined Mobile Applications* field, and select the appropriate mobile application. To remove the selection and view all items for all mobile applications in the system, click in the field again and select the asterisk (*) symbol.

Creating, Copying, Deleting, and Changing Items

The following standard actions are available to configure different components and items within your mobile application setup:

- **Create:** Creates a new item. All modifiable fields are empty.
- **Copy:** Copies the item that was highlighted and creates a new item. All modifiable fields are filled in with the information from the existing item and are available for changes before saving.
- **Delete:** Deletes the highlighted item.
- **Change:** Allows you to change the highlighted item in the modifiable fields.

Saving or Canceling Changes for an Item

Once you click the *Create*, *Copy*, or *Change* button, the *Save* and *Cancel* buttons are displayed. After you change the configuration of the item, click *Save* to save the changes or *Cancel* to discard the changes.

> **i** Note
>
> If the *Save* and *Cancel* buttons are active, the *Home* link for the ConfigPanel is not available. Either save your changes or cancel out of the changes to return to the main Configuration Panel page

Message List

Certain actions can generate system messages. These messages can be error messages or informational messages. If you perform an action that prompts a system message, a message bar appears above the main panel with a brief description of the message.

Click the *Show List* button to display the detailed view of the message list.

2.2.2 Mobile Application Configuration

The Mobile Application Configuration page allows you to configure general settings for the entire mobile application.

The *Mobile Application Configuration* page contains the following tabs:

- General
- Mobile Status Setting
- Conversion Exit Setting (not used in SAP Asset Manager)
- System Components (not used in SAP Asset Manager)
- Parameters
- Client Globals (not used in SAP Asset Manager)
- User Attributes (not used in SAP Asset Manager)

General Tab

Use the *General* tab to create or change basic information about a mobile application.

- **Basic Data section:** Enter the name of the mobile application in the `<Mobile Application>` field, which is limited to 40 characters. Select the type of application in the `<Type>` field. Note that for SAP Asset Manager, the type is `<oData Applications>`. Enter a brief, easy to understand description in the `<Description>` field, limited to 60 characters. Type in the release number of the application in the `<Release>` field.
- **User Management Setting:** When the `<Disable Automatic User Creation>` box is checked, a new user GUID is not automatically created when a new mobile client is detected in the system. Manually create and maintain mobile users through the Administration portal.
- **Server Management Setting:** When the `<Disable Automatic Server Registration>` box is checked, a new server GUID is not automatically created when a new server is detected in the system. You must manually create and maintain servers through the Administration portal.

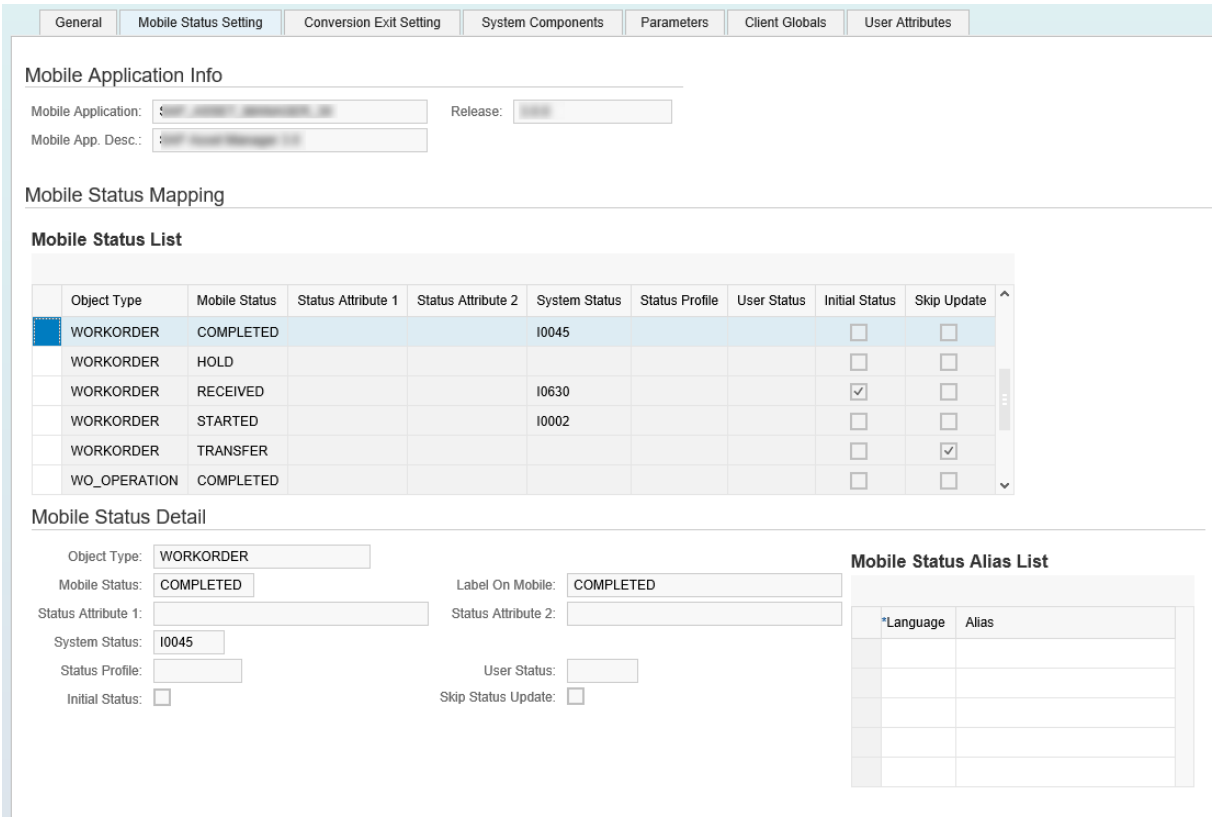
- **Life-cycle management:** When the <Application Blocked> box is checked, the mobile application is disabled. The mobile user can no longer connect to the back-end system for the mobile application, and the xChange process is also disabled. The <Effective Date> and <Time> fields flag when the change takes effect.
- **xChange Setting:** When the <Disable Change Detection> box is checked, the change detection process, or xchange process, for the application is completely disabled.
- **Inbound Transaction Management:** Not used in the SAP Asset Manager application.
- **Multi Backend Setting:** When checked, enables a specific mobile application to connect to multiple SAP systems, consisting of one host server and one or more satellite servers.
- **System Role:** Dropdown menu where you can select either *Host* or *Satellite*.
  A *Host* system is the connection between SAP and the SAP Asset Manager application in the SAP Cloud Platform. The host server provides the logic to the application and functions as the bridge to the satellite server or servers. There can only be one host server per system.
  Satellite servers communicate with SAP through the host server. To complete multi backend configuration, configure the host and back-end servers using the *System Components* tab. See that section for more details.

Mobile Status Setting Tab

Use the Mobile Status Setting tab to map the available mobile statuses that an oData mobile data object (OMDO) supports on the client side. If a user status also exists for the same object type, you can link it to the mobile status and the system status through this tab.

- **Mobile Application Info:** The <Mobile Application> field is read only and is the name of the mobile application. The <Mobile Application Description> is read only and is a brief description of the mobile application. The <Release> field is read only and is the release number of the application.
- **Mobile Status Mapping:** Use the <Add Status> and <Delete Status> buttons to create and delete mobile status mappings. Fill out the <Object Type> with the specific object in the mobile application, for example, <Notification>. The <Mobile Status> is the status defined by the mobile application. The <Label on Mobile> is not used. The <User Status> is an SAP status code as defined in SAP. Note that the status codes are language independent codes.
  If the <Initial Status> checkbox is selected, the mobile status is displayed by default when you download the object to the mobile device. To skip a specific mobile status update from a mobile device, use the <Skip Update> checkbox corresponding to the mobile status object.
  Use the *Mobile Status Alias List* table to define language-specific mobile status aliases.

In the following example screen, the highlighted row in the mapping table indicates that if a user sets a work order to completed, the application sets the work order system status to I0045 in SAP.



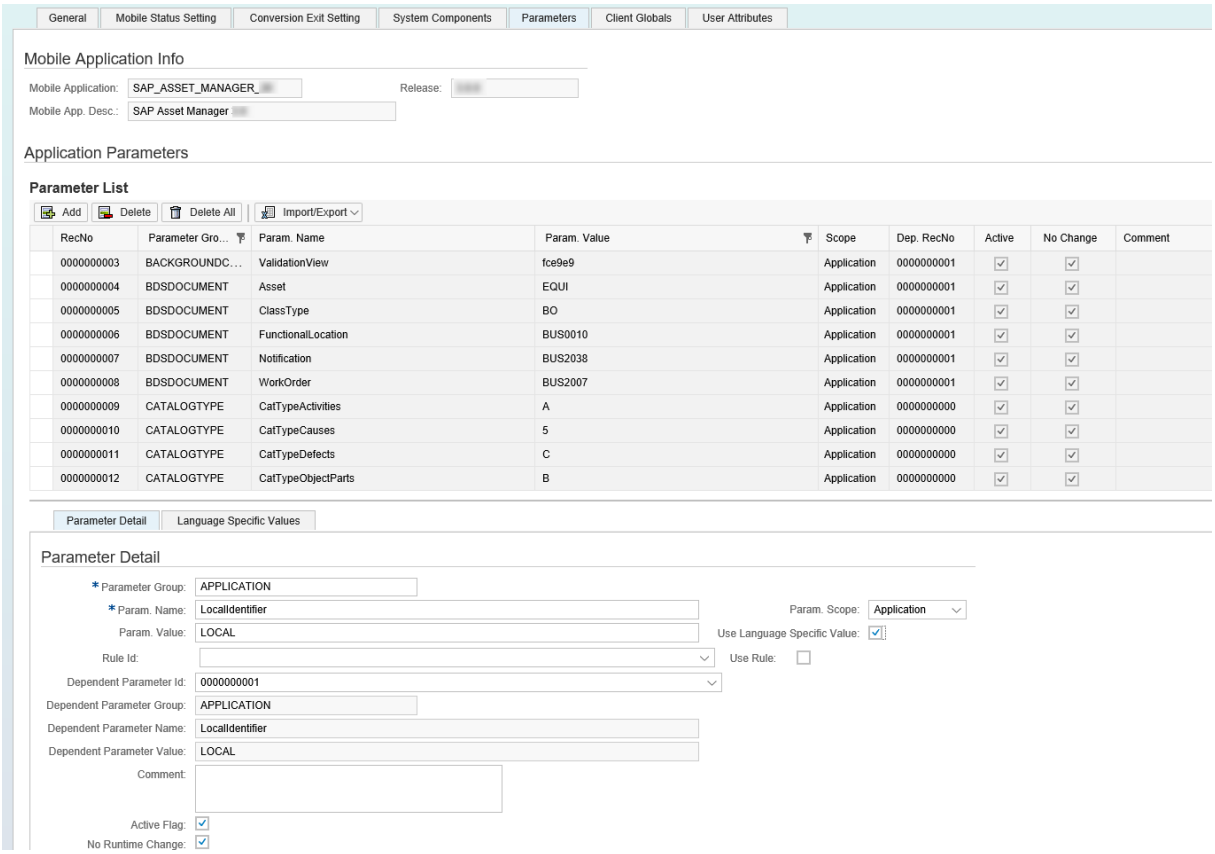
If there is no system status or user status, the mobile status only affects the mobile device and does not affect the backend SAP system.

If there is a user status specified but no status profile when the mobile user sets the mobile status, the app sets that user status for the object, disregarding the status profile of that object.

If there is a user status and status profile specified when the mobile user sets the mobile status, the app sets that user status if the object uses that status profile.

Parameters Tab

The *Parameters* tab defines system parameters.



- **Mobile Application Info:** The `<Mobile Application>` field is read only and is the name of the mobile application. The `<Mobile Application Description>` is read only and is a brief description of the mobile application. The `<Release>` field is read only and is the release number of the application.
- **Application Parameters:** Use the `<Add>` and `<Delete>` buttons to create and delete parameters.
- **Parameter Detail:** The `<Parameter Group>` is the group to which the parameter belongs. Groups are how you organize parameters. References to a parameter include both the group name and the parameter name. The `<Parameter Name>` is the unique name of the parameter.
  The `<Parameter Value>` is the currently configured value of the parameter. References to the parameter return the configured value. Use the `<Language Specific Value>` checkbox to select which parameters you wish to be language dependent. The checkbox and the corresponding *Language Specific Values* tab are only active after you have clicked the *Change* button. Note that the language available in the *Language Specific Values tab*, *Value List* table, depends on the language you are using to log into the mobile client. You must manually maintain each parameter that you wish to control for a language specific value separately.
  Set the `<Parameter Scope>` to one of the following options:
  - Mobile Application: Value for all users of the application
  - Mobile User: Value you can override for individual users. To override a parameter value of a user, see the Administration & Monitoring Portal on parameters

> **i Note**
>
> For information on setting user parameters, see the following security guides, depending on your back end system:
>
> - Mobile Add-On for S/4HANA Security Guide

- Mobile Add-On for ERP Security Guide

The <Rule ID> field contains the rule used at runtime. If you check the <Use Rule> box, the rule in the <Rule ID> field is active.
Check the <Active Flag> box to ensure that the parameter is used by the mobile application. Inactive parameters are not used by the application. When you check the <No Runtime Change> box, you cannot override the value of the parameter. The configured value is always the value. If the box is not checked, the parameter values can be overridden at runtime through synchronization processing.

2.2.3 Geospatial Service Definitions

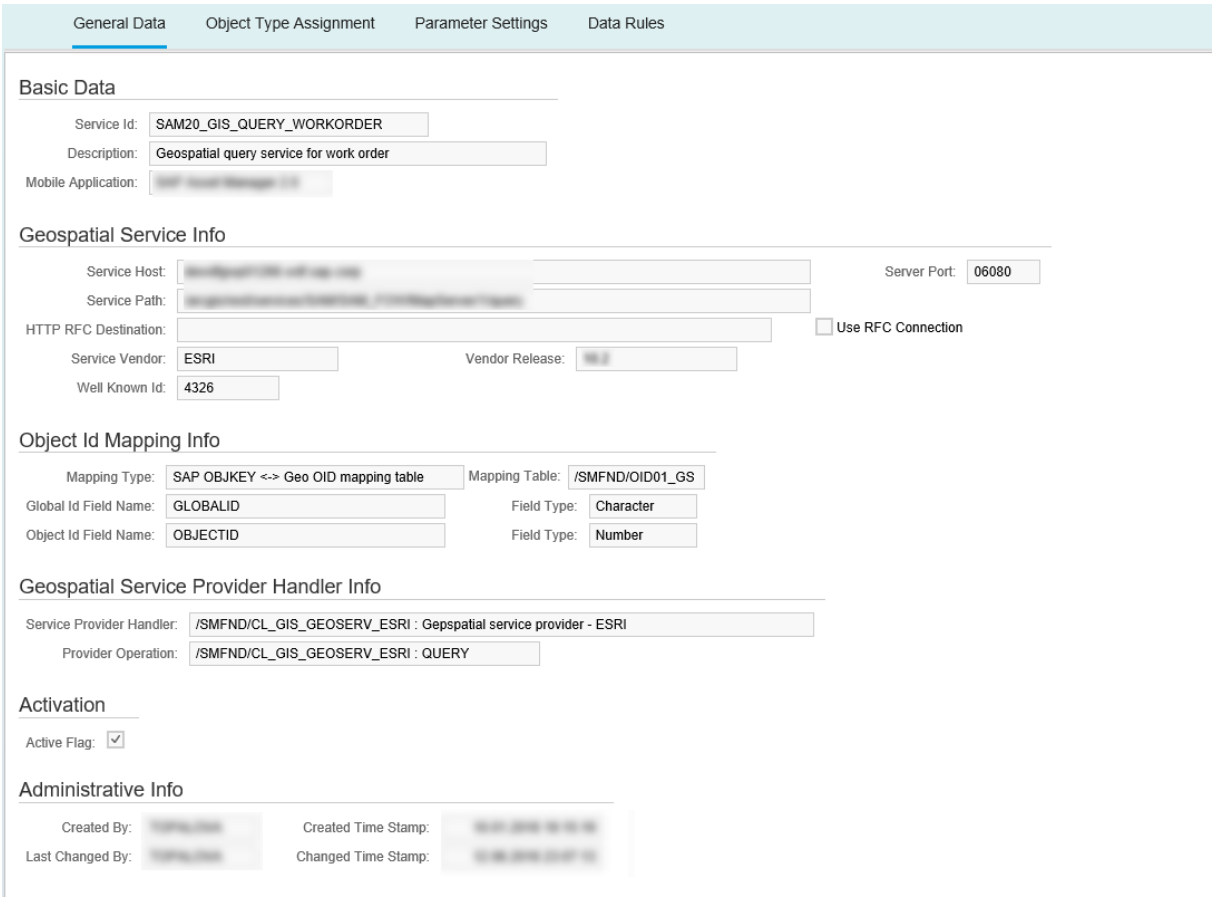
A geographic information system (GIS) integrates hardware, software, and data for capturing, managing, analyzing, and displaying all forms of geographically referenced information.

Geospatial data plays an important part in the daily operations of many organizations. By adding geospatial data to the technical data of an asset, you get a full picture of that asset.

The *Geospatial Service Definitions* page contains the following tabs:

- General Data
- Object Type Assignment
- Parameter Settings
- Data Rules

General Data Tab



General Data tab:

Basic Data

- **Service ID:** Required field. Name of the geospatial service ID, limited to 40 characters, with namespace protection. Use the Y or Z namespace.
- **Description:** Description of the geospatial service
- **Mobile Application:** Mobile application of the geospatial service. Every geospatial service is assigned to a specific mobile application.

Geospatial Service Info

- **Service Host:** Host name of the geospatial service provider
- **Server Port:** Port number of the geospatial service provider
- **Service Path:** End-point URL of the geospatial service
- **HTTP RFC Destination:** RFC destination pointing to the GIS. By using the RFC destination, you can support various logon authentication methods of the geospatial service. If you use an RFC destination, service host, port, and path are not required.
- **Use RFC Destination:** Indicator that you must use an RFC destination to connect to GIS
- **Service Vendor:** Vendor name of the GIS provider.SAP uses Esri for its out of the box installation of the SAP Asset Manager application.

- **Vendor Release** Version information of the GIS provider. The Esri version for SAP Asset Manager 1911 is *ArcGIS iOS SDK 100.2.1* for iOS and *ArcGIS Runtime SDK for Android 100.3.0* for Android.
- **Well Known ID:** Well-known ID of the coordinate system

Object ID Mapping Info

- **Mapping Type:** Defines how the SAP object ID is mapped to the object ID in the geospatial service provider database. Select from the following:
  - **No mapping:** SAP object ID is the same as the object ID in the GIS provider database
  - **SAP OBJKEY <-> GEO OID-mapping table:** Mapping is stored in a mapping table
  - **Lookup Routing:** Use a lookup ABAP object-oriented class to determine the mapping dynamically. The ABAP object-oriented lookup class must implement the `/SMFND/IF_GIS_OID_MAPPING` interface.
- **Mapping Table:** SAP mapping table name, if used. The standard `/SMFND/OID01_GS` mapping table is the default. The standard `/SMFND/GIS_OIDMAP_UPLOAD_PROG` program is provided to load the table with a text file
- **Object ID Lookup Routine:** OID lookup routine name, if used. Used in conjunction with the *Lookup Routing* mapping type
- **Global ID Field Name:** Name of the Global ID field in the GIS provider database, if relevant. For example, in the Esri database, the global field name is *GLOBALID*
- **Object ID Field Name:** Name of the `<Object ID>` field in the GIS provider database, if relevant. For example, in the Esri database, the field name is *OBJECTID*
- **Field Type** Use the dropdown to select the type of field for both the `<Global ID>` and the `<Object ID>`
- **Service Provider Handler:** ABAP object-oriented class that handles integration to the GIS provider
- **Provider Operation:** ABAP OO class operation that handles integration to the GIS service provider.

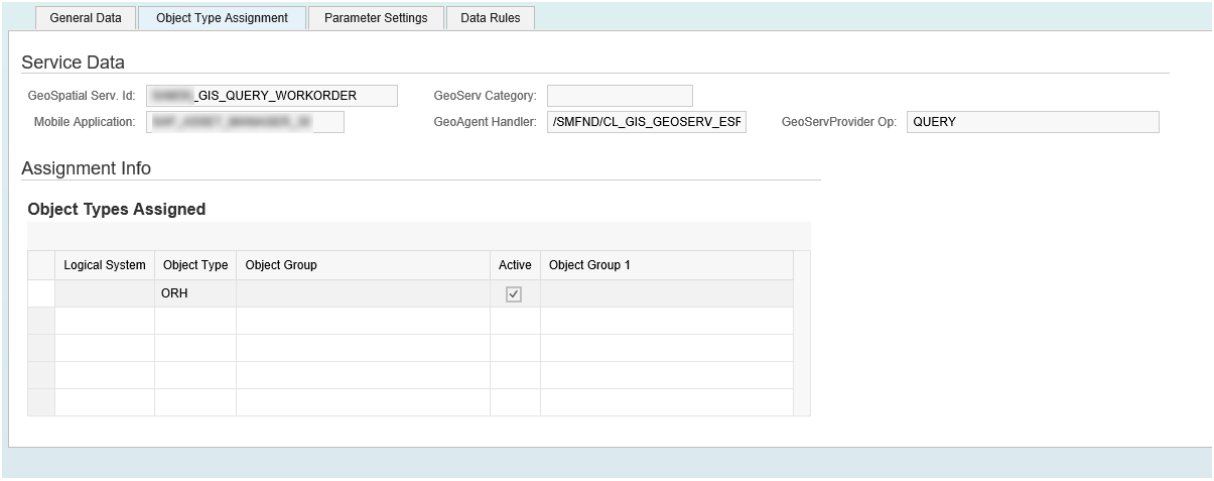
Activation

- **Active Flag** Check the checkbox to activate the GIS query service

Object Type Assignment Tab

Use the *Object Type Assignment* tab to define what type of SAP objects are assigned to the geospatial service. You can define different geospatial services for different SAP object types.

For example, you can map equipment with polygon geospatial data to one geospatial feature layer. You can then map equipment with point geospatial data to a different geospatial feature layer.

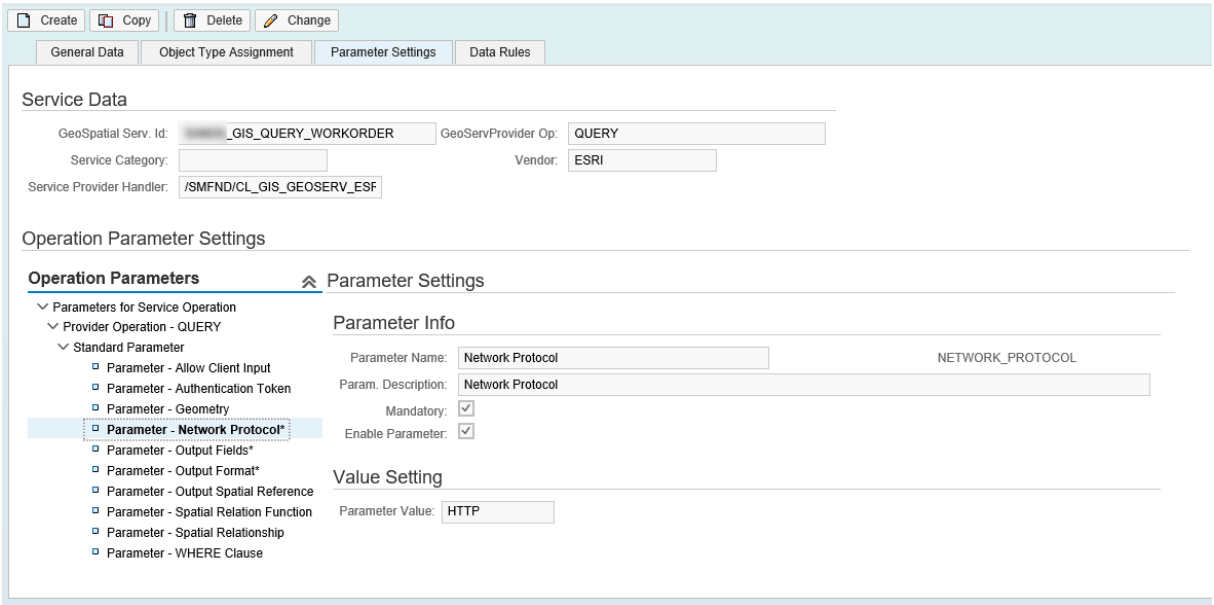


Assignment Info

- **Logical System:** Logical system of the SAP object. A logical system is required to properly identify the SAP object if the mobile add-on aggregates data from different back-end systems.You can configure the following fields on the
- **Object Type:** Type of the object as it is identified in the mobile add-on. For example, the standard object type *IEQ* is used to identify the *Equipment* object.
- **Object Group:** Optional setting used to further group the objects in the same object type.
- **Active:** When the *Active* checkbox is marked, the assignment is active.
- **Object Group 1:** Optional setting used to further group objects of the same object type and object group.

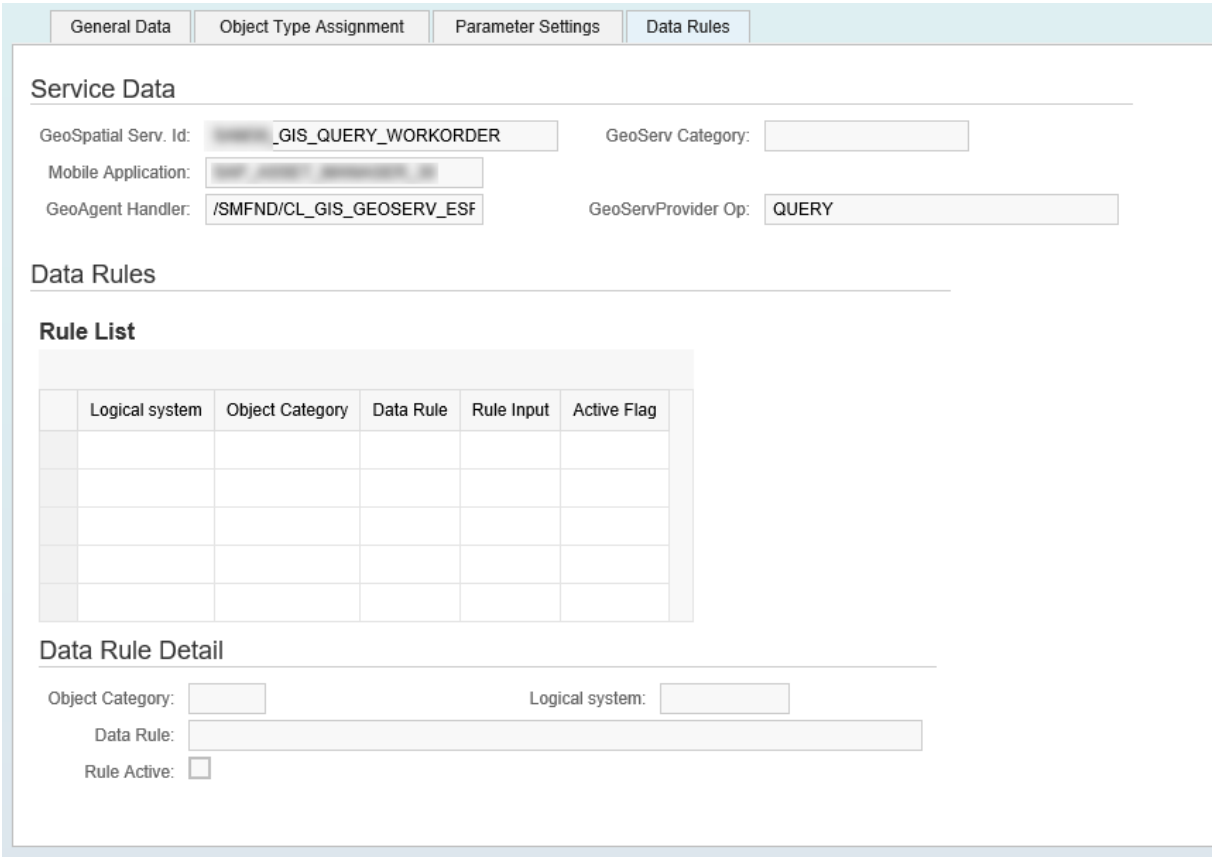
Parameter Settings Tab

Use the *Parameter Settings* tab to define parameter settings for the service provider handler. The service provider handler can declare the list of parameters that might require input. If parameters are declared, they are displayed on this tab, and you can enter values for them.



Data Rules Tab

Use the *Data Rules* tab to define data rules. A data rule is used to transform input data to the service provider handler, before calling the geospatial service. For example, to dynamically assign values of object type, object group, and object group 1 to input data, use a data rule. Using a data rule influences which geospatial service is assigned to an input object.



Data Rule Detail

- **Object Category:** Type of the object as it is identified in the mobile add-on. For example, you can use the standard object type *IEQ* to identify the *Equipment* object.
- **Logical System:** Logical system of the SAP object. If the mobile add-on aggregates data from different back-end systems, the logical system is required to identify the object.
- **Data Rule:** Data rules are ABAP OO classes that implement the `/SMFND/IF_GIS_DATA_RULE` interface.
- **Rule Active:** If the checkbox is checked, the rule is active.

2.2.4 OData Channel Integration Settings

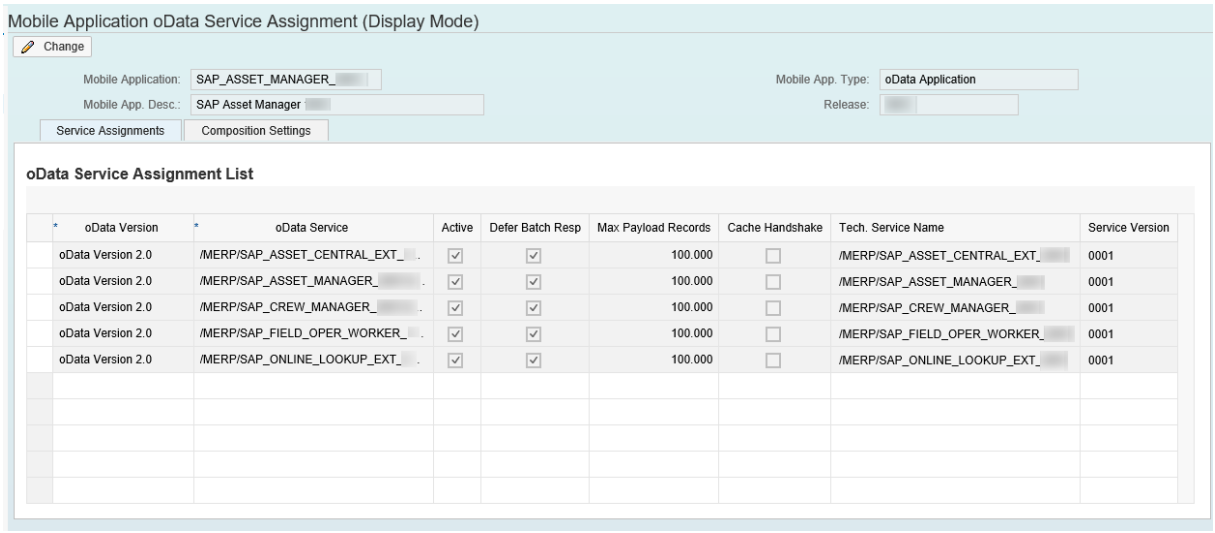
2.2.4.1 OData Service Assignment

Gateway OData services implemented using the Mobile Integration Framework for SAP are different from the typical Gateway OData services.

The following requirements must be met for the Gateway OData services:

- Define the Gateway OData technical model using the generic model provider class of the Mobile Integration Framework `/MFND/CL_CORE_ODATA_V2_MPC`. You can maintain the OData technical model with transaction `/IWBEP/REG_MODEL`.
- Define the Gateway OData technical service using the generic data provider class of the Mobile Integration Framework `/MFND/CL_CORE_ODATA_V2_DPC`. You can maintain the OData technical service with transaction `/IWBEP/REG_SERVICE`.
- Assign the Gateway OData technical service to a mobile application by choosing the *OData Service Assignment* in the ConfigPanel.
- Do not define the Gateway OData technical model using the Gateway Service Builder. The model is determined and generated dynamically by the generic model provider class `/MFND/CL_CORE_ODATA_V2_MPC` based on the model configuration settings defined in the ConfigPanel.
- The generic data provider class `/MFND/CL_CORE_ODATA_V2_DPC` doesn't provide the required business logic for the Gateway OData technical service. Business logic is provided by OMDOs. Assign every OData business request to the service to an OMDO. The assigned OMDO performs the necessary business logic for the business request.

Service Assignments



You can define the following settings for the OData service assignment:

- **OData Version:** OData version 2.0 is supported.
- **OData Service:** The Gateway OData technical service that is assigned to the mobile application. You can only assign a single mobile application to a Gateway OData technical service. Assignment to multiple mobile applications is not supported.
- **Active:** If the *Active* checkbox is checked, both the OData model and OData service are supported. If the checkbox is not checked, no OData model is generated for the service, and the data request to the service is not supported.
- **Defer Batch Response:** This setting is only relevant if you have an SAP S/4HANA back-end system. If the checkbox is checked, the Gateway runtime deferred batch response is enabled, which can improve performance during read request processing.

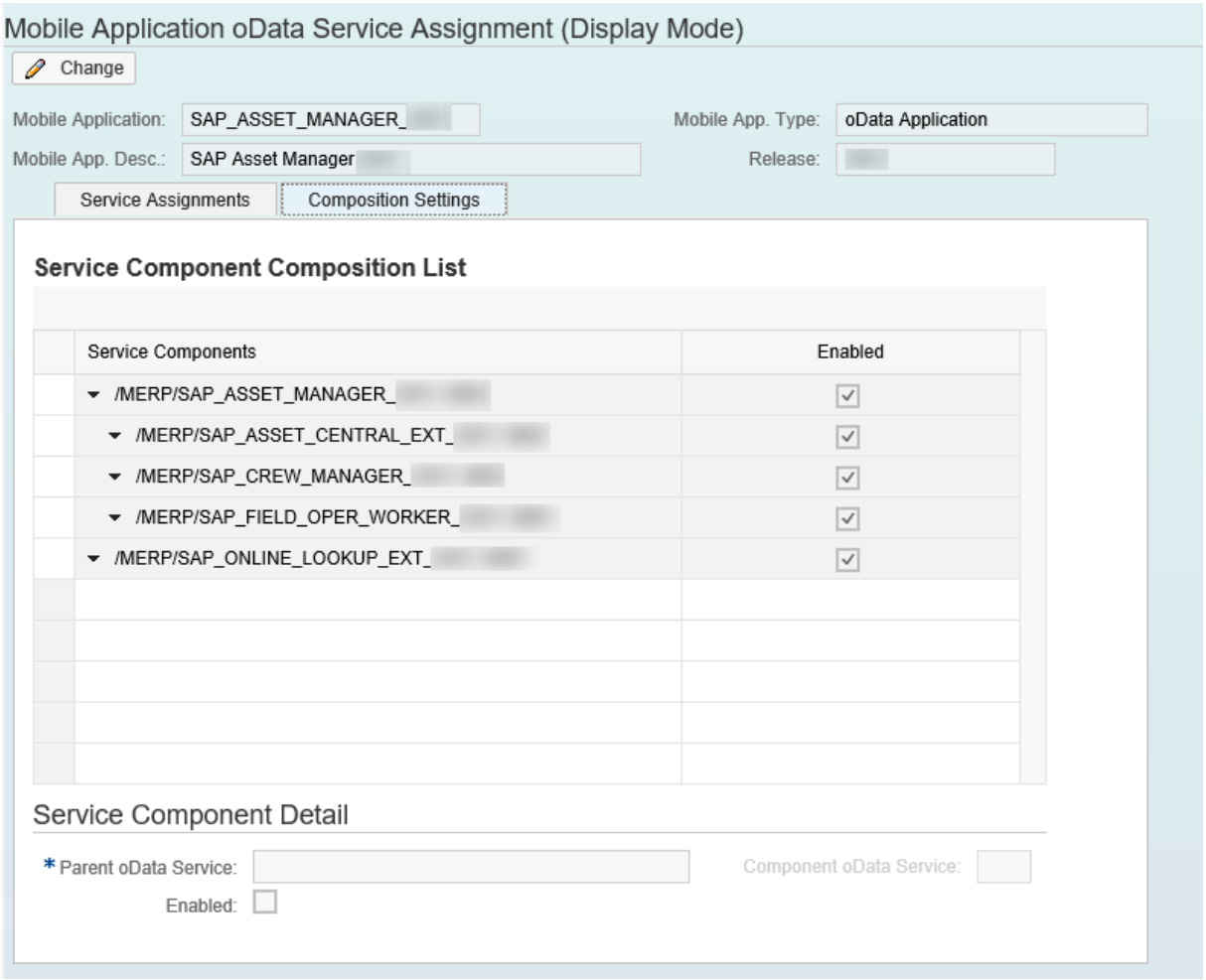
- **Max Payload Records:** Maximum number of records in the response payload to a read request. In case the read request is a batch request with multiple read requests, this setting defines the total number of records allowed in the overall payload, which is the aggregation of individual read request responses.
- **Cache Handshake:** When checked, the Gateway runtime metadata cache handshake is enabled for the service.
- **Technical Service Name:** Read-only information. Gateway OData technical service name.
- **Service Version:** Read-only information. Gateway OData technical service version.

Composition Settings

With service component composition, you can compose a complex service using component services.

In the following example, service `/MERP/SAP_ASSET_MANAGER_1911 version 0001` is composed of service `/MERP/SAP_ASSET_CENTRAL_EXT_1911 version 0001`, service `/MERP/SAP_CREW_MANAGER_1911 version 0001`, and service `/MERP/SAP_FIELD_OPER_WORKER_1911 version 0001`.

The entity model for service `/MERP/SAP_ASSET_MANAGER_1911 version 0001` includes the entity model from `/MERP/SAP_CREW_MANAGER_1911 version 0001` and the other two entity models listed. The entity model for service `/MERP/SAP_CREW_MANAGER_1911 version 0001` only contains its own entities.



To define a component composition, define the following:

- **Parent OData Service and Version:** Parent OData service. Entity model of a child OData service is included in the parent entity model. Association and navigation properties can be defined between parent service and component service.
- **Component OData Service and Version:** Child OData service
- **Enabled:** If the checkbox is not checked, the entity model of the component service is not included in the entity model of the parent service.

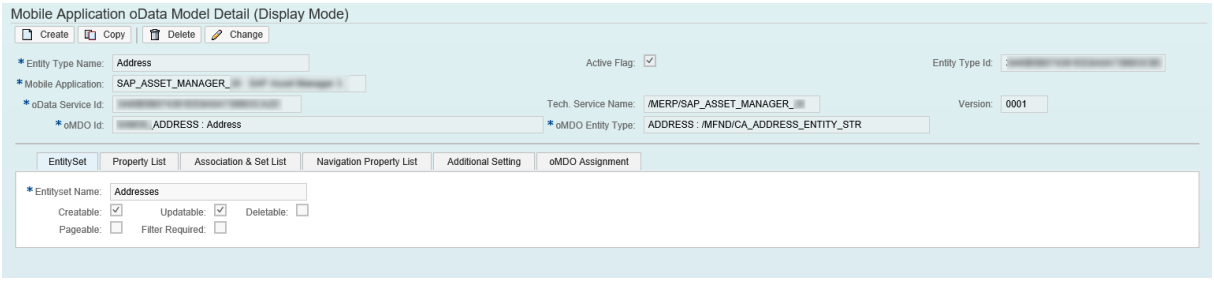
2.2.4.2 OData Model Configuration

OData service implemented using the Mobile Application Integration Framework does not use the Gateway Service Builder to define the OData model. Define the OData model using the OData Model configuration tool in the ConfigPanel. The runtime OData model is generated dynamically based on the configuration settings. The OData model configuration is mobile application-specific. You cannot share OData models across mobile applications.

Define the OData model configuration settings through the following screens:

Entity Set Tab

Entity configuration defines the OData entity type. Entity set configuration defines the OData entity set. In an OData model configuration, each entity type is limited to one entity set. Reuse of entity types by multiple entity sets or by different OData services is not supported.



The following attributes are available for the *Entity Type* definition:

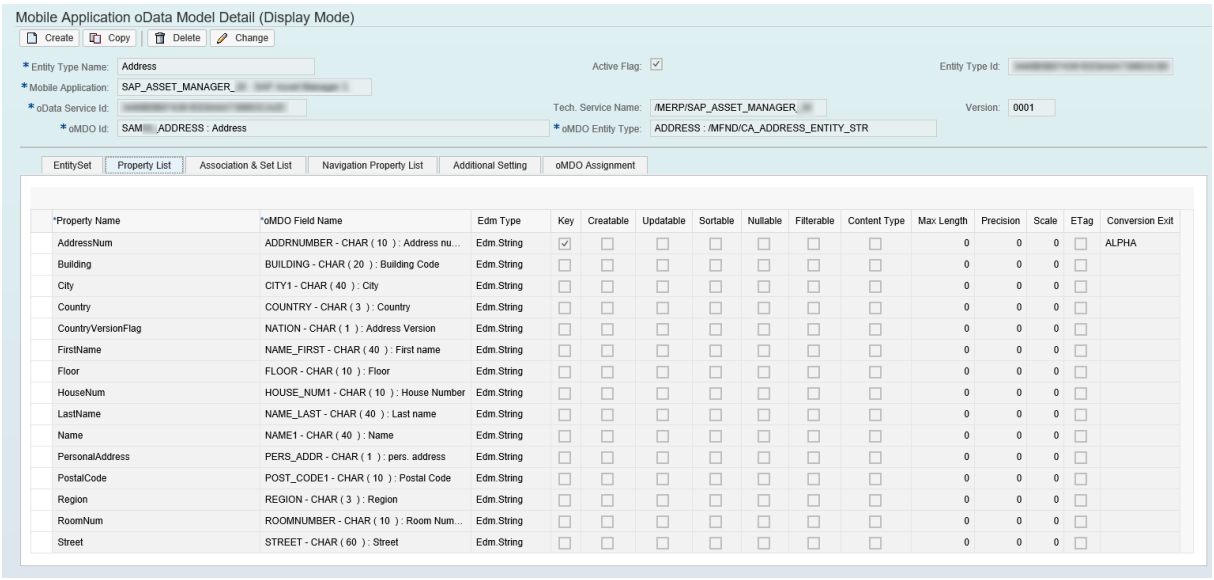
- **Entity Type Name:** Case-sensitive name of the entity type. The name must be unique within the OData service.
- **Active Flag:** If unchecked, the entity type is not included in the generated OData model
- **Entity Type ID:** Internal ID generated by the system to identify the entity type
- **Mobile Application:** Mobile application for the entity type. The OData model configuration is defined for individual mobile applications. You can reuse the entity type name in different mobile applications.
- **Internal OData Service ID:** Internal OData service ID that identifies the OData service for which the entity type is defined
- **Service:** Gateway technical service name of the OData service. Information is read-only.
- **Version:** Gateway technical service version. Information is read-only.

- **OMDO ID:** OMDO that provides business logic for the entity type and its entity set
- **OMDO Entity Type:** Technical entity type of the OMDO that is mapped to the OData entity type. Data for the OData entity type is supplied by the OMDO entity type.

The following attributes are available for the *Entity Set* definition:

- **EntitySet Name:** Case-sensitive name of the entity set. Must be unique within the OData service.
- **Creatable:** If checked, creation (POST) request for the entity set is supported
- **Updatable:** If checked, update (PUT / PATCH / MERGE) request for the entity set is supported
- **Deletable:** If checked, deletion (DELETE) request for the entity set is supported
- **Pageable:** If checked, paging is allowed for the entity set read request
- **Filter Required:** Not applicable for SAP Asset Manager

Property List



*Property Name	*oMDO Field Name	Edm Type	Key	Creatable	Updatable	Sortable	Nullable	Filterable	Content Type	Max Length	Precision	Scale	ETag	Conversion Exit
AddressNum	ADDRNUMBER - CHAR (10) : Address nu...	Edm.String	☑	☐	☐	☐	☐	☐	☐	0	0	0	☐	ALPHA
Building	BUILDING - CHAR (20) : Building Code	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
City	CITY1 - CHAR (40) : City	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
Country	COUNTRY - CHAR (3) : Country	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
CountryVersionFlag	NATION - CHAR (1) : Address Version	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
FirstName	NAME_FIRST - CHAR (40) : First name	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
Floor	FLOOR - CHAR (10) : Floor	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
HouseNum	HOUSE_NUM1 - CHAR (10) : House Number	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
LastName	NAME_LAST - CHAR (40) : Last name	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
Name	NAME1 - CHAR (40) : Name	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
PersonalAddress	PERS_ADDR - CHAR (1) : pers. address	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
PostalCode	POST_CODE1 - CHAR (10) : Postal Code	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
Region	REGION - CHAR (3) : Region	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
RoomNum	ROOMNUMBER - CHAR (10) : Room Num...	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	
Street	STREET - CHAR (60) : Street	Edm.String	☐	☐	☐	☐	☐	☐	☐	0	0	0	☐	

You can configure the following on the *Property List* tab:

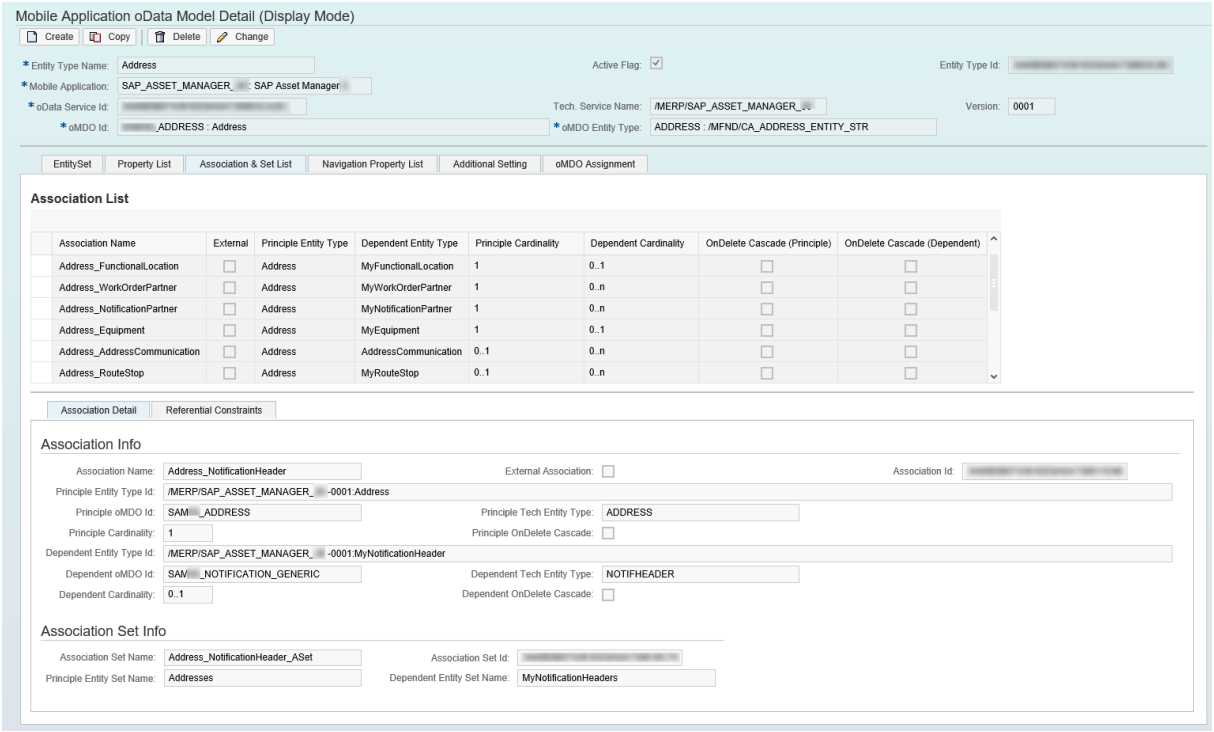
- **Property Name:** Case-sensitive name of the property
- **OMDO Field Name:** Field name of the OMDO technical entity type mapped to the property. The value of the property is supplied in the `<OMDO Entity Type>` field.
- **EDM Type:** Standard EDM type for the property
- **Key Indicator:** If checked, the property is the key property of the entity type
- **Creatable:** Not applicable for SAP Asset Manager
- **Updatable:** Not applicable for SAP Asset Manager
- **Sortable:** If checked, the property can be used for sorting
- **Nullable:** If checked, the property can have NULL value
- **Filterable:** If checked, the property can be used for filtering
- **Content Type:** If checked, the property value can represent content type
- **Max Length:** Maximum allowed length of the property value

- **Precision:** Decimal precision of the property, if relevant
- **Scale:** Decimal scale of the property, if relevant
- **ETag:** If checked, the property serves as the `<ETag>` field. You can set only one field as an `<ETag>` field in an entity type.
- **Conversion Exit:** Conversion exit for the property

Association & Set List Tab

An association defines the relationship between two entity types, with one entity type as the principle entity type, and the other as the dependent entity type. An association set defines the relationship between the two entity sets of the respective entity types in the association. In an OData model configuration, associations and association sets are child objects of an entity type, and each association can have only one association set defined.

When you define an OData model to use with OData offline SDK client application, you also define referential constraints for the association. Only key fields of the principle entity type can be used in referential constraints.



You can configure the following in the *Association Info* section:

- **Association Name:** Case-sensitive name of the association
- **External Association:** By default, the parent entity type of the association is the principle entity type. However, if the *External Association* flag is checked, the parent entity type of the association is the dependent entity type. An external association is commonly used when defining associations between entity types across component services.
- **Association ID:** A read-only internal GUID generated by the system that identifies the association
- **Principle Entity Type ID:** Lead entity type of the association

- **Principle OMDO ID:** Read-only OMDO ID to which the principle entity type is mapped to
- **Principle Tech Entity Type:** Read-only technical entity type of the OMDO ID to which the principle entity type is mapped to
- **Principle Cardinality:** Cardinality of the lead entity type
- **OnDelete Cascade (Principle):** If checked, the dependent entity type and entity set are automatically deleted when the principle entity type and entity set are deleted
- **Dependent Entity Type ID:** For standard associations, the dependent entity type can be any entity type belonging to the same OData service or child component service
- **Dependent OMDO ID:** Read-only OMDO ID to which the dependent entity type is mapped to
- **Dependent Tech Entity Type:** Read-only technical entity type of the OMDO ID to which the dependent entity type is mapped to
- **Dependent Cardinality:** Cardinality of the dependent entity type
- **Dependent OnDelete Cascade:** If checked, the principle entity type and entity set are automatically deleted when the dependent entity type and entity set are deleted

You can configure the following in the *Association Set Info* section:

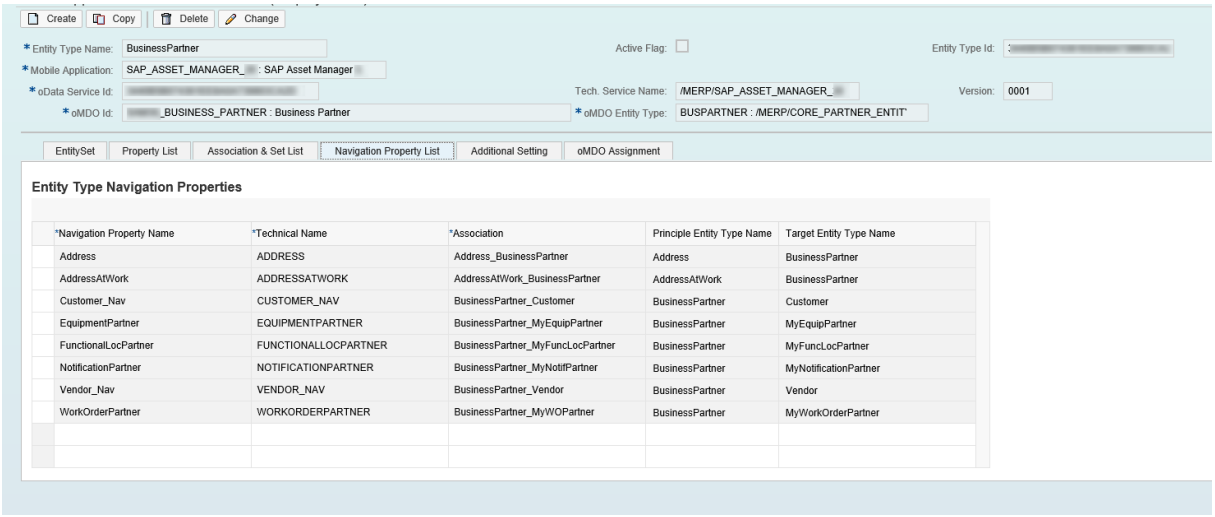
- **Association Set Name:** Case-sensitive name of the association set
- **Association Set ID:** Read-only internal GUID generated by the system that identifies the association set
- **Principle Entity Set Name:** Read-only entity set name of the principle entity type. Each entity type in the OData model configuration can only have one entity set.
- **Dependent Entity Set Name:** Read-only entity set name of the dependent entity type. Each entity type in the OData model configuration can only have one entity set

You can configure the following in the *Referential Constraints* section (not pictured in detail in the example screenshot):

- **Principle Entity Type:** Read-only principle entity type of the association
- **Principle Property:** Principle property of the referential constraint. The principle property is the key field of the principle entity type.
- **Dependent Entity Type:** Read-only dependent entity type of the association
- **Dependent Property:** Dependent property of the referential constraint. The dependent property can be any field of the dependent entity type that has a foreign key relationship with the principle property.

Navigation Property List Tab

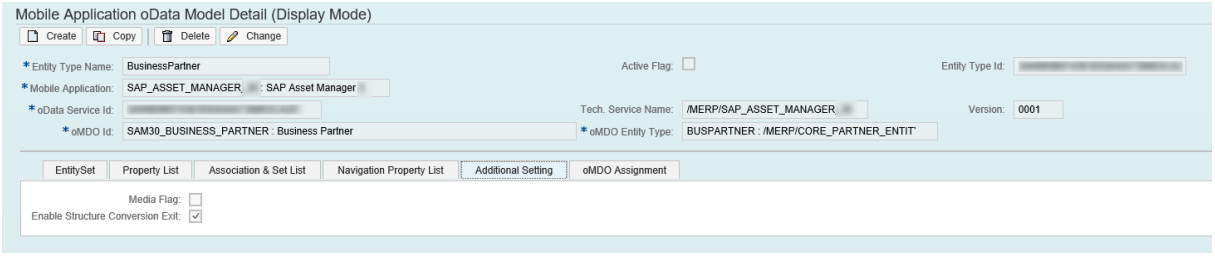
A navigation property represents a link from the parent entity type to a related entity types.



You can define the following attributes for a navigation property in the *Entity Type Navigation Properties* table:

- **Navigation Property Name:**
- **Technical Name:** Case-sensitive name of the Internal technical name of the navigation property. The technical name is not case-sensitive.
- **Association:** Association for the navigation property. The navigation represents the link between the principle entity type and the dependent entity type.
- **Principle Entity Type Name:** Read-only principle entity type name of the association used by the navigation property
- **Dependent Entity Type Name:** Read-only dependent entity type name of the association used by the navigation property

Additional Setting Tab



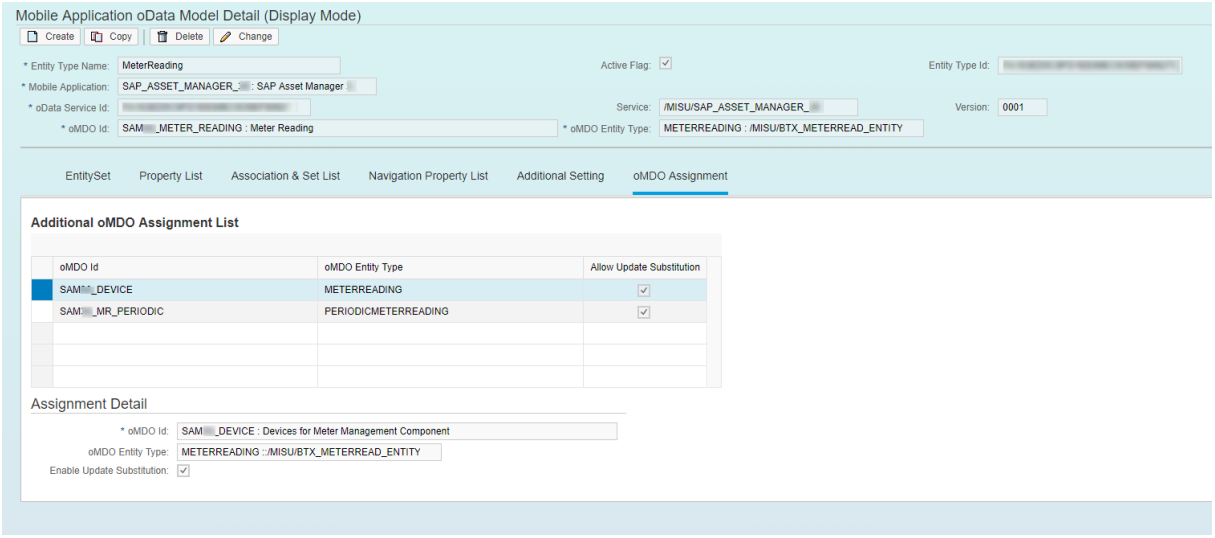
You can define the following additional settings for the OData model:

- **Media Flag:** If checked, the entity type is a media entity type
- **Enable Structure Conversion Exit:** If checked, the gateway automatically performs conversion exits for the entity type at runtime for both the request payload and the response payload

OMDO Assignment Tab

The *OMDO Assignment* tab only supports substitution configuration scenarios.

Use the following screenshot as an example. When a user posts a meter reading from their client, by default the reading is posted to the default OMDO, which here is *SAM<XX>_METER_READING*. However, if the user is reading a periodic meter, the reading is posted to the *SAM<XX>_MR_PERIODIC* OMDO, which is substituted for the default OMDO through the use of custom headers.

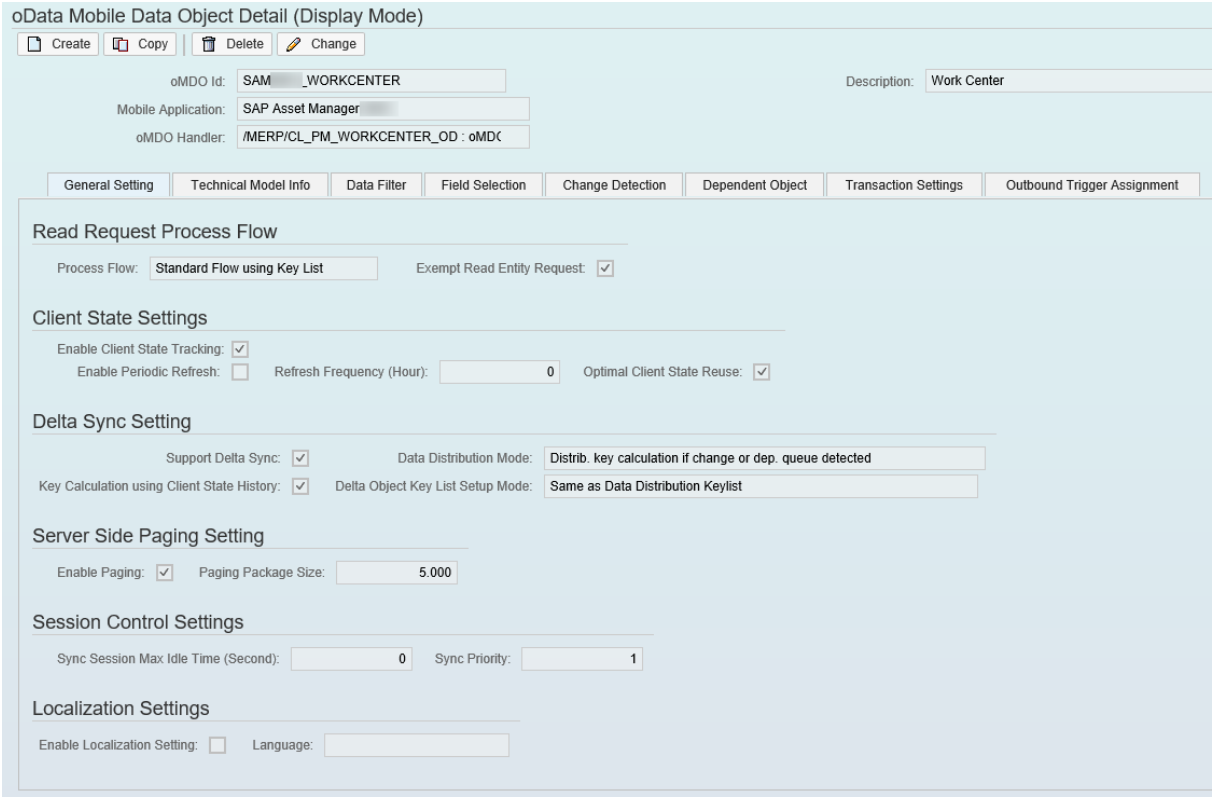


2.2.4.3 OData Mobile Data Object Configuration

An OData mobile data object (also known as OMDO) provides business logic for a business object used in an OData-based mobile application. An OMDO provides both technical implementation and configuration support for the represented business object, including all aspects of related operations such as object creation, update, deletion, or read and downloading. The OMDO also supports configuration such as data distribution rules for data download.

OData requests for a business object are mapped to an OMDO object. The OMDO handler then processes the requests for the OMDO object. For read requests, the OMDO handler considers and enforces the data distribution rules and other configuration settings, and determines the proper output response. For create, update, and delete requests, the OMDO handler creates or updates the SAP BusinessObjects in the back-end system as requested in the OData requests, and provides the relevant response.

General Setting Tab



You can set the following attributes on the *General Setting* tab:

- **OMDO ID:** ID of OData Mobile Data Object; limited to 40 characters. The OMDO ID must be unique in an SAP client, across all mobile applications, as namespace restriction is enforced. A customer-defined OMDO ID must use the Y or Z namespace.
- **Description:** Short description of the OMDO, limited to 60 characters
- **Mobile Application:** Mobile application of the OMDO. An OMDO always belongs to a single mobile application.
- **OMDO Handler:** An ABAP OO class that provides the technical implementation for the OMDO object. The OMDO handler must be a subclass of `/MFND/CL_CORE_OMDO_HNDLER_BASE`. You can reuse an OMDO handler to provide technical implementation for multiple OMDO objects.
- **Process Flow:** Determines how the OMDO handler processes OData entity set read requests. Based on the process flow setting, different OMDO handler methods are invoked at runtime. The OMDO handler determines which process flow it supports.
  - **Standard Flow using Key List:** Typically used by the OMDO handler to support complex SAP BusinessObjects with multiple entity types, complex relationships between entity types and data distribution rules, and with change detection support. Examples of SAP BusinessObjects that use the standard flow include transaction data objects such as work orders or notifications; and master data objects such as equipment or functional locations.
  - **Basic Flow without Key List:** Typically used by the OMDO handler to support simple SAP BusinessObjects with a single entity type and no change detection support. Examples of SAP BusinessObjects that use the basic flow without key list include customizing data, such as plant or order type. OMDO handlers that only support basic flow are typically subclasses from `/MFND/CL_CORE_OMDO_BASIC_HNDLR`.

- **Exempt Read Entity Request:** If checked, the entity read request is handled by the *READ_ENTITY_REQUEST_PROC* method. The method supports on-demand entity read requests without interference with client state tracking for entity set read requests.
- **Enable Client State Tracking:** If checked, the client state records are maintained for each entity set read request. Client state tracking enablement is required for other features such as key list calculation using client state history, periodic refresh support, and optimized client state reuse.
    - **Client State:** Captures the list of calculated object keys that are sent to the mobile client for the entity set read requests, the data distribution rules used for the calculation, and the time of the calculation. By enabling client state management, the system has a record of the objects distributed to the mobile client. Enabling client state management also allows the system to calculate the list of objects to be removed from mobile client via tombstones.
      At runtime, the client state info generated during client synchronization can be displayed using the client state monitor through the Administration & Monitoring Portal.
- **Enable Periodic Refresh:** If checked, every entity set delta sync read request is checked if it qualifies for periodic refresh. A periodic refresh means that if time between an entity set initial sync or last period refresh read request and current read request has exceeded the defined frequency, all objects from the client are removed and a new object key list is calculated based on the current data distribution rules.
- **Refresh Frequency (Hour):** Defines the periodic frequency refresh in number of hours
- **Optimal Client State Reuse:** If checked, the system tries to reuse previous client state records whenever possible instead of generating new client state records. Reuse of previous client state records can improve runtime performance. Whether a client state record can be reused depends on the object key list and data distribution rules. If both items remain unchanged compared to the previous client state, the client state record can be reused.
- **Support Delta Sync:** If checked, a delta token is generated and returned in response to the entity set read request
- **Key Calculation using Client State History:** This setting is effective only if client state tracking is enabled. If checked, the object key list is taken from current data distribution calculation, and the list is compared against the object key list from the previous client state. Also, objects that should be removed from the client are automatically identified, and downloaded to the client again.
- **Data Distribution Mode:** Controls when data distribution key calculation is performed for delta sync entity set read requests. It's only relevant if OMDO handler supports the *Standard Flow using Key List* process flow. There's overhead associated with data distribution key calculation. You can reduce the number of key calculations during sync to improve performance. However, the potential performance gain has to be measured against the correctness of the response. If the object key list from data distribution changes frequently, skipping the calculation can lead to incorrect results.
    - **Always perform distribution key calculation:** Perform the data distribution key calculation for every read request
    - **Distribution key calculation if change or dependent queue detected:** Perform data distribution key calculation only if there's data change detected in the back-end system, or if there are new dependent queue entries inserted
    - **No distribution key calculation. Delta calculation only:** Always skip data distribution key calculation in delta sync
- **Delta Object Key List Set up Mode:** Used in standard flow to determine how the delta object key list is initialized. The delta object key list determines the list of objects that are included in the delta sync response.
    - **Same as Data Distribution Key List:** Default delta sync object key list is the same as the data distribution key list. If change detection is supported by the OMDO, we recommend using this setting.

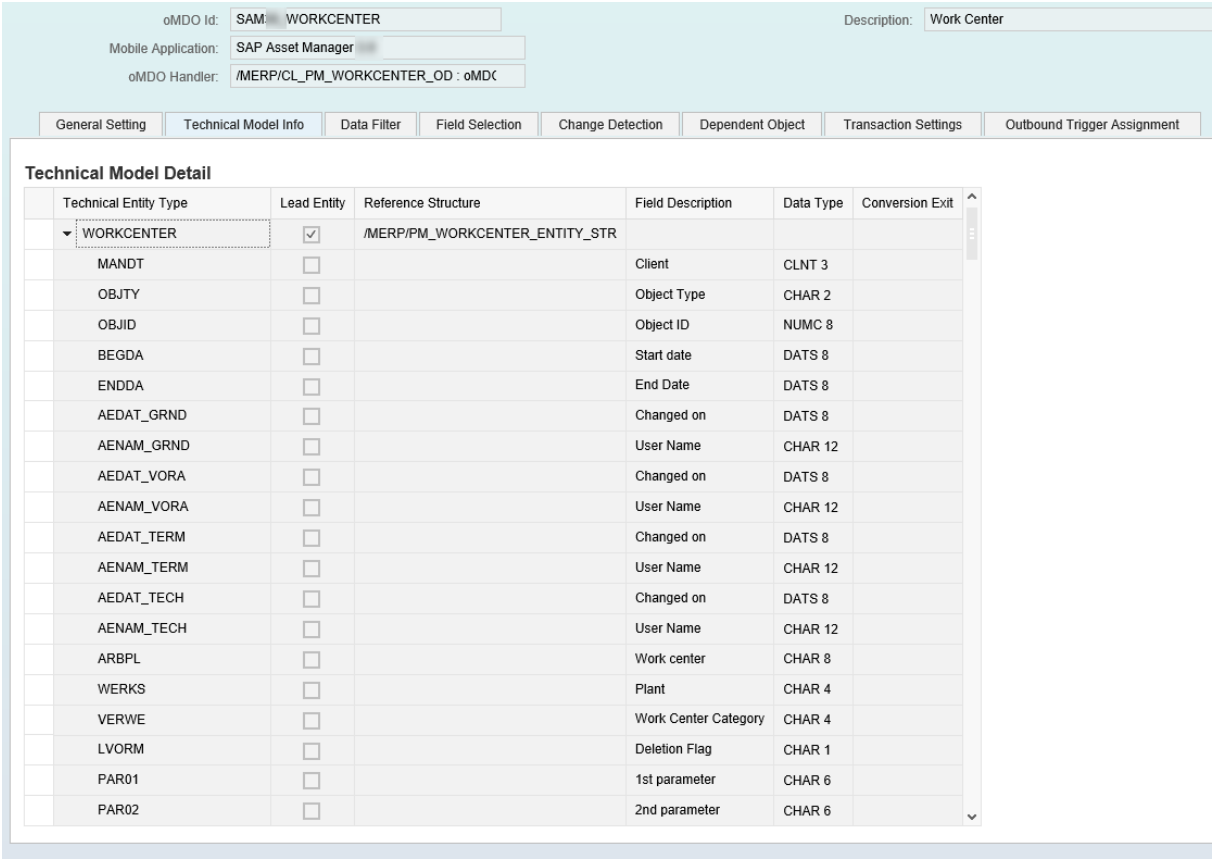
Remove objects that do not have changes since the last delta sync from the delta object key list to prevent unnecessary download and to improve performance.

  - **Difference of Current and Last Data Distribution Key List:** The current data distribution key list is compared to the object key list from the last client state. The only objects that are not included in the last client state object list are included in the delta sync object key list. If no change detection is supported by the OMDO, we recommend using this setting.

- **Enable Paging:** If checked, server-side paging is activated. When paging is active, response from the OMDO to the OData request is limited to the specified paging package size. If the number of records in the response exceeds the paging package size, the response is divided into separate pages. The *$skiptoken* identifies the next page that is generated and included in the response to the client. The client then sends a follow-up request with the skip token to retrieve the next page. This process continues until the client retrieves all pages.
- **Paging Package Size:** Maximum number of records that can be included in a page for the response of the OMDO. For read requests in a batch, the maximum payload records number defined for the OData service assignment is also considered. If the overall number of records in the batch response has reached the maximum payload records number, the number of records in the individual request response can be less than the paging package size specified.
- **Sync Session Max Idle Time (Second):** Estimated maximum duration of a sync session for a single OMDO. For requests belonging to the same OMDO, the response can be calculated beforehand for all requests once and then reused. When the requests are received separately, as long as the requests are received within the defined sync time duration, the response calculated beforehand can be used, instead of calculating it again. Calculating a response beforehand can improve performance. If the sync session max idle time is set to *0*, each request triggers the calculation for the response.
- **Sync Priority:** Represents the processing sequence for read requests in a batch, where *0* has the first priority sequence. For OMDO objects with the same sync priority, OData read requests for these OMDO objects are processed based on the sequence in the batch.
- **Enable Localization Setting:** When checked, brings data associated with the OMDO fetch translated to the language specified in the `<Language>` field irrespective of what logon language is used for the connection. All other OMDOs that don't have *Enable Localization Setting* enabled use the logon language.
- **Language:** Used when the `<Enable Localization Setting>` box is checked. Select your desired language from the dropdown list.

Technical Model Info Tab

The *Technical Model Info* tab is a display only tab. This tab displays the technical entity model supported by the OMDO handler.

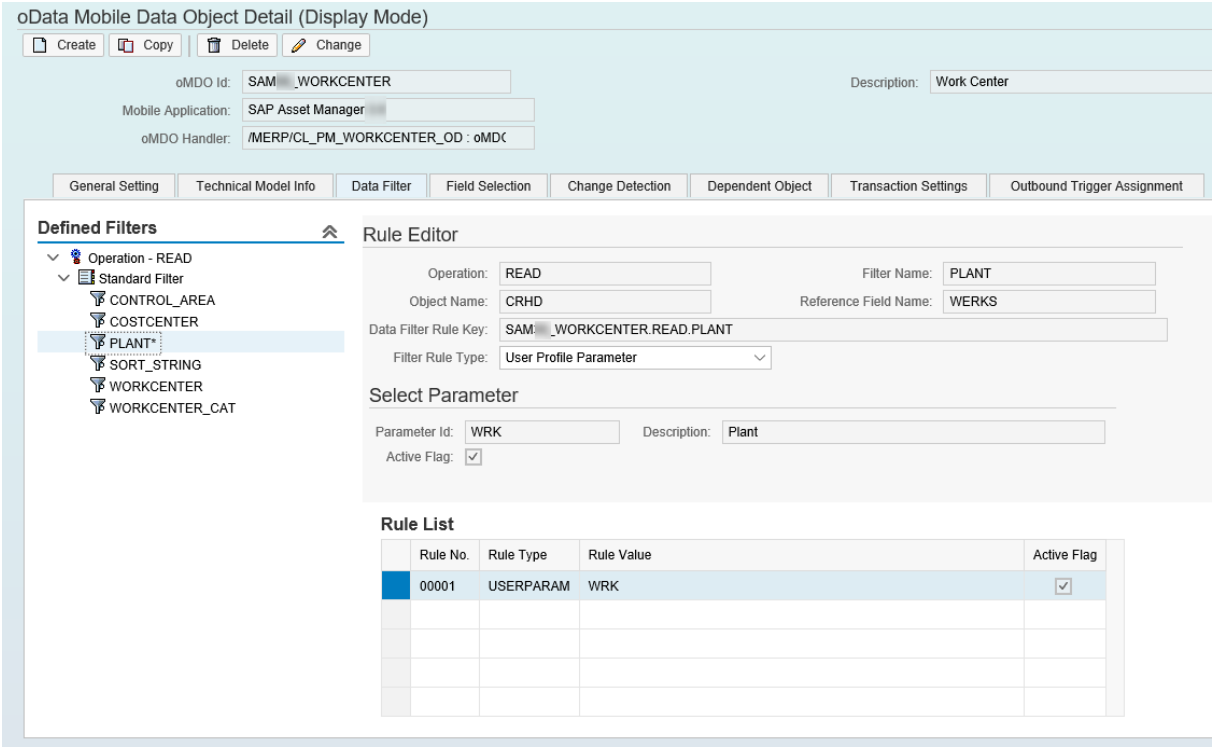


Technical Entity Type	Lead Entity	Reference Structure	Field Description	Data Type	Conversion Exit
WORKCENTER	☑	/MERP/PM_WORKCENTER_ENTITY_STR			
MANDT	☐		Client	CLNT 3	
OBJTY	☐		Object Type	CHAR 2	
OBJID	☐		Object ID	NUMC 8	
BEGDA	☐		Start date	DATS 8	
ENDDA	☐		End Date	DATS 8	
AEDAT_GRND	☐		Changed on	DATS 8	
AENAM_GRND	☐		User Name	CHAR 12	
AEDAT_VORA	☐		Changed on	DATS 8	
AENAM_VORA	☐		User Name	CHAR 12	
AEDAT_TERM	☐		Changed on	DATS 8	
AENAM_TERM	☐		User Name	CHAR 12	
AEDAT_TECH	☐		Changed on	DATS 8	
AENAM_TECH	☐		User Name	CHAR 12	
ARBPL	☐		Work center	CHAR 8	
WERKS	☐		Plant	CHAR 4	
VERWE	☐		Work Center Category	CHAR 4	
LVORM	☐		Deletion Flag	CHAR 1	
PAR01	☐		1st parameter	CHAR 6	
PAR02	☐		2nd parameter	CHAR 6	

- **Technical Entity Type:** Technical entity type that the OMDO handler supports
- **Lead Entity:** Indicates whether a technical entity is the lead entity type supported by the OMDO handler. The lead entity type represents the header record of a business object. An OMDO operates on a business object level. For an OMDO CREATE operation, a create request (POST request) for the lead entity type is required. If the lead entity already exists, a CREATE request (POST request) for nonlead entity types are considered as OMDO UPDATE operations.
- **Reference Structure:** Data dictionary structure of the technical entity type
- **Field Name:** Field name from the data dictionary structure
- **Field Description:** Field description
- **Data Type:** Field data type
- **Conversion Exit:** Assigned conversion exit for the field

Data Filter Tab

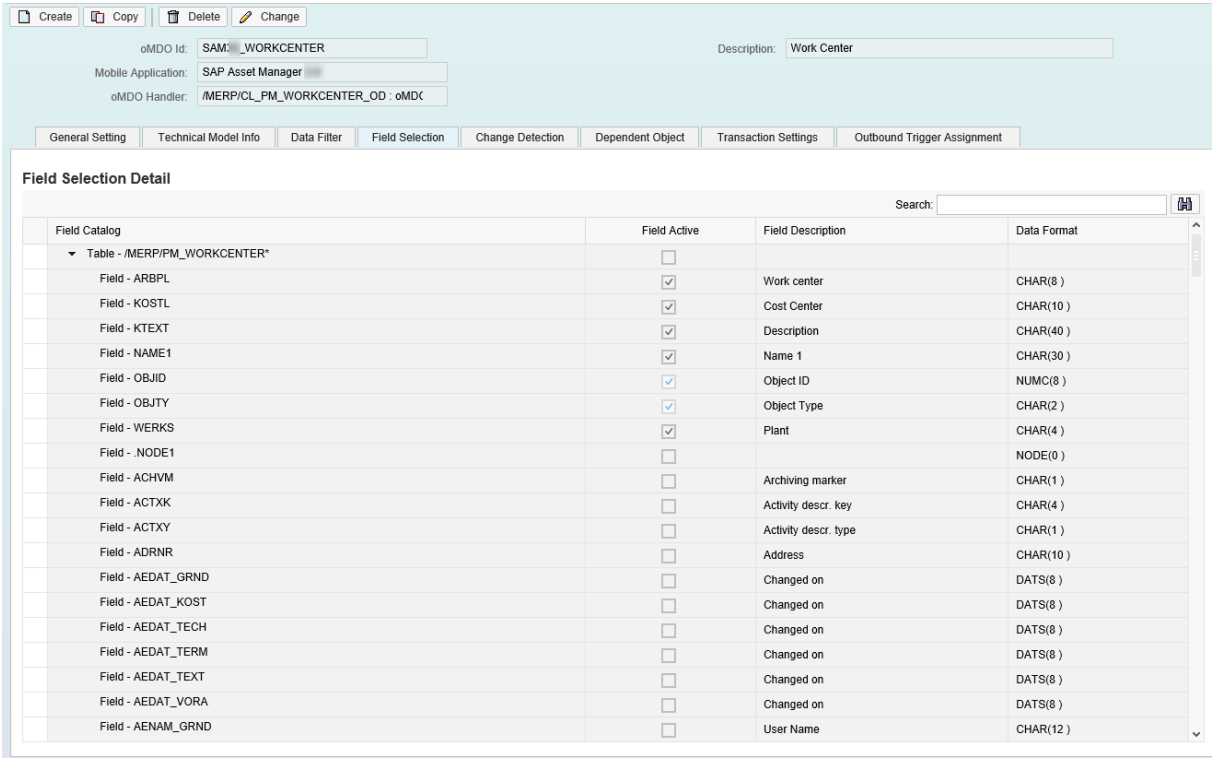
An OMDO handler can declare data filters and parameters supported by its CRUD (CREATE / READ / UPDATE / DELETE) operations. These filters are displayed on the *Data Filter* tab.



- **Defined Filters:** The *Defined Filters* navigation area displays the list of supported filters for the OMDO, grouped by operation and filter group. There are two types of filters available, though both types may not be available for every OMDO:
  - **Standard Filter:** A standard filter is a single field filter that references a table or structure field that is displayed in the `<Object Name>` field and the `<Reference Field Name>` field
  - **Table Filter:** A table filter is a structured filter that references a table or structure that is displayed in the `<Object Name>` field.
- **Rule Editor:** Details for a filter are displayed in this section when you select a filter from the *Defined Filters* section.
- **Filter Rule Type:** You can select one of the four types of rules to define:
  - **Static Value in Range Table Format:** Static rule, with the rule value defined at design time in the configuration
  - **User Profile Parameter:** Dynamic rule, with the rule value evaluated at runtime based on the runtime profile parameter value of the user. You can display and maintain the user profile parameter value with the transaction code *SU3*.
  - **Mobile User Attribute:** Dynamic rule, with the rule value evaluated at runtime based on the runtime mobile user attribute of the user. You can display and maintain the mobile user attribute in the Administration & Monitoring Portal, which you can access with the transaction */SYCLO/ADMIN*. Then select *Administration* > *User Management*.
  - **Filter Handler:** Dynamic rule, with the rule value evaluated at runtime by a filter handler. A filter handler is an ABAP subclass of `/MFND/CL_CORE_OMDO_RULE_BASE`.

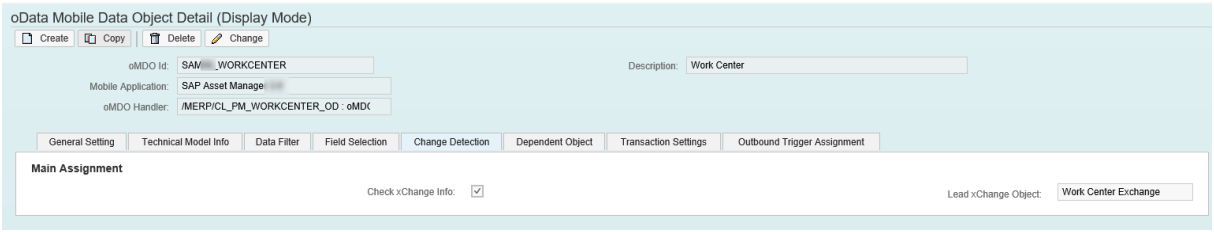
Field Selection Tab

An OMDO handler can declare field catalogs supported for the READ operation. If there is a READ operation, by default, all of the fields from the database tables related to the OMDO object are selected. Using the field catalog, customers can control which fields are selected, and improve performance, as typically a mobile application doesn't require all of the fields.



Change Detection Tab

You can enable change detection for the OMDO using the *Change Detection* tab.



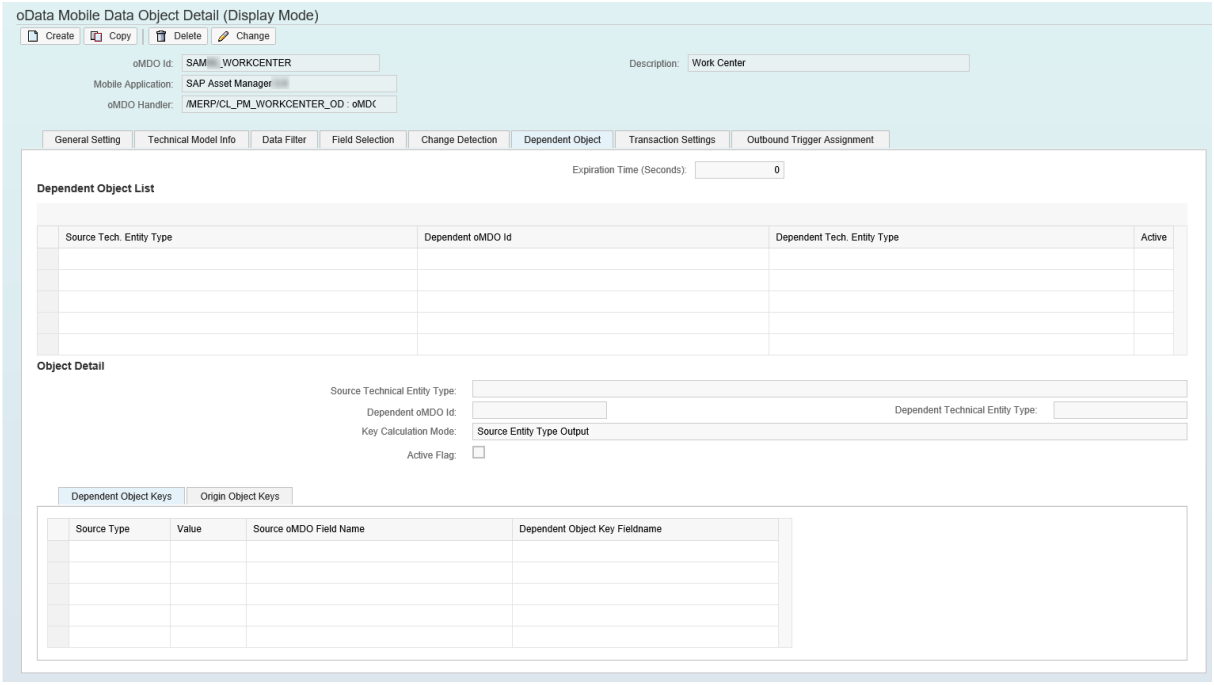
- **Check xChange Info:** Applies to standard flow processing only. If checked, change detection info is checked to determine the delta sync object key list.
- **Lead xChange Object:** xChange object that supplies the change detection information for the OMDO. Information from the xChange table of the xChange object is read and used for the calculation of the delta sync object key list.

Dependent Object Tab

Define settings related to dependent objects on the *Dependent Objects* tab.

In some business cases, the read request sequence for the OMDOs or SAP BusinessObjects is important, since the data distribution object key list of a subsequent OMDO depends on the results or outputs of the precedent OMDOs. The subsequent OMDO is treated as a dependent object of the precedent OMDO. The leading OMDO is the source OMDO, as the output of the lead OMDO supplies information for the dependent OMDO. Dependent object key information generated by the leading OMDO is stored in the dependent object queue, and is used by the dependent OMDO during its read request processing.

For example, SAP Asset Manager downloads detail information for equipment and functional locations used in work orders assigned to a technician. To fulfill this requirement, read requests for work order assignments occur first, and equipment and functional locations are set up as dependent objects for the work order OMDO.



You can define the following settings for a dependent object of the current OMDO:

- **Source Technical Entity Type:** Source OMDO technical entity type that contains information required by the dependent object
- **Dependent OMDO ID:** ID of the dependent OMDO
- **Dependent Technical Entity Type:** Receiving technical entity type of the dependent OMDO, for which information from the source technical entity type is transferred
- **Key Calculation Mode:** Select the way the keys are passed to the OMDO. Key calculation is a dependent object concept; how you set up your dependent object is based on your source object.
    - Source Entity Output: Input for the dependent key. Keys are calculated based on the source entity type output.
    - Source Entity Type Distribution Key List: Dependent Object Key construction comes from the distribution key list of the source entity type. Using this option always collects all the valid keys from the source entity type.

    - Source Entity Type Output + Target Entity Type Client State: Similar to Source Entity Output plus the previous client state of the target entity type. Here, what is being created for dependent object collection is a combined collection of the source entity type output and the target entity type client state records from the previous sync.
- **Active Flag:** Enable or disable a dependent object definition

You can define the following settings for the mapping info of dependent object keys in the *Dependent Object Keys* tab:

- **Source Type:** Use option *By Field Name* if the information comes from a field of the source technical entity type. Use option *By Value* if a constant value is used.
- **Source Value:** Constant value for a dependent object key field. This field is only relevant if the source type is set to *By Value*.
- **Source OMDO Field Name:** Name of the source technical entity type field that supplies value for the dependent object key. This field is only relevant if the source type is set to *By Field Name*.
- **Dependent Object Key Field Name:** Field name of the dependent technical entity type that receives the value from the source technical entity type field

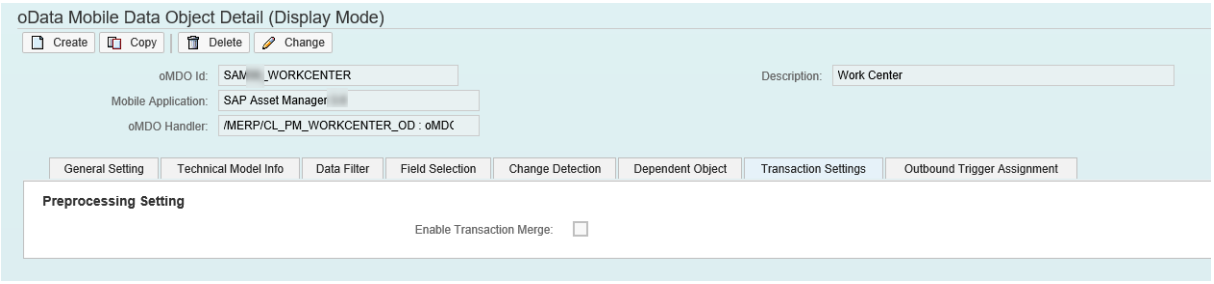
You can define the following settings for the mapping info of origin object keys in the *Origin Object Keys* tab (not shown in detail in the example screenshot). The origin object key identifies the source OMDO object that has generated the dependent object key.

- **Source Type:** Use option *By Field Name* if the information comes from a field of the source technical entity type. Use option *By Value* if a constant value is used.
- **Source Value:** Constant value for an origin object key field. This field is only relevant if the source type is set to *By Value*.
- **Source OMDO Field Name:** Name of the source technical entity type field that supplies value for the origin object key. This field is only relevant if the source type is set to *By Field Name*.

You can display the dependent object queues generated during client synchronization at runtime using the *Dependent Queue Monitor* on the Administration & Monitoring Portal.

Transaction Settings Tab

You can define settings related to transactions (CUD requests) on the *Transaction Settings* tab.



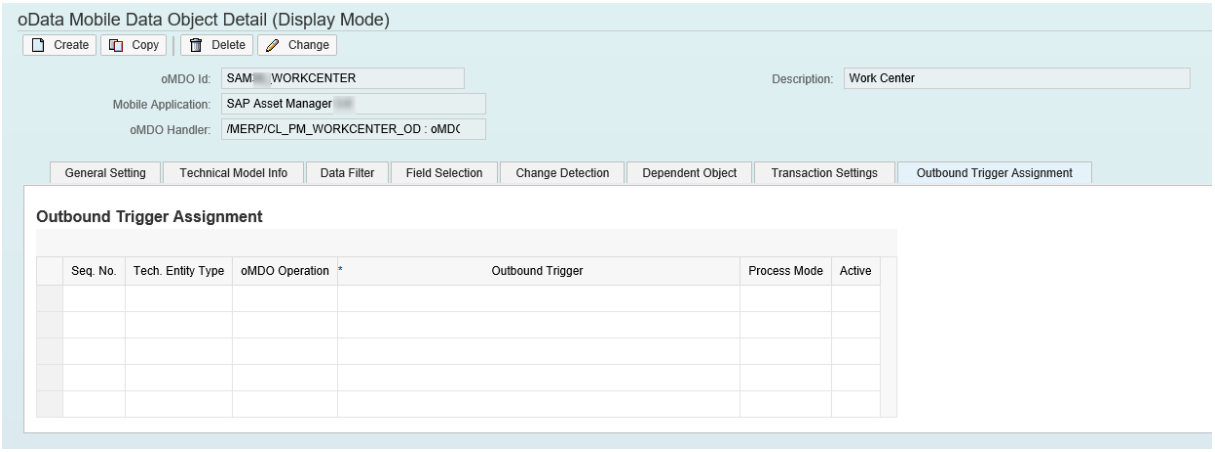
- **Enable Transaction Merge:** If checked, transaction requests for the same object that are received in the same changeset are merged. Therefore, the number of requests processed by the OMDO handler is reduced. The sequence of the transaction requests in the changeset is respected, with the attribute value of the last transaction request as the final value for the attribute.

For example, for *Object 123* the requests are as follows:

Requests in Changeset	Requests Processed by OMDO
Request #1 CREATE 123 Request #2 UPDATE 123 Request #3 DELETE 123	None
Request #1 CREATE 123 Request #2 UPDATE 123 Request #3 UPDATE 123	Request #1 CREATE 123 (attribute values from Request #2 and Request #3 are merged into Request #1)
Request #1 UPDATE 123 Request #2 UPDATE 123	Request #1 UPDATE 123 (attribute values from Request #3 merged into Request #1)

Outbound Trigger Assignment

An outbound trigger performs a function that is implemented by the outbound trigger handler. Outbound triggers can be assigned to an OMDO. The assigned outbound triggers are invoked after OMDO processing has been completed, based on the sequence of the assignment.



You can set the following attributes when assigning an outbound trigger to an OMDO:

- **Technical Entity Type:** Optional. If defined, the outbound trigger is invoked only if the specified technical entity type was processed by the OMDO.
- **OMDO Operation:** Optional. If defined, the outbound trigger is invoked only if the specified OMDO operation is processed.
- **Outbound Trigger ID:** Assigned outbound trigger ID
- **Process Mode:** Only the *Always Run* mode is supported
- **Active:** Enable or disable an outbound trigger

2.2.5 Change Detection Settings

Change detection settings are used to define and configure how the mobile application, such as SAP Asset Manager, communicates with SAP and the object tables contained within SAP

The following areas are used to configure backend change detection:

- **Exchange Object Configuration:** Change detection rules for SAP data objects, such as master data and transaction data, defined for each mobile application
- **EFI Assignment:** Enhancement framework implementation trigger assigned to exchange objects

> **i** Note
>
> Create tables and objects in SAP and the Mobile Development Kit before you can create or configure information in the ConfigPanel.

2.2.5.1 EFI Assignment

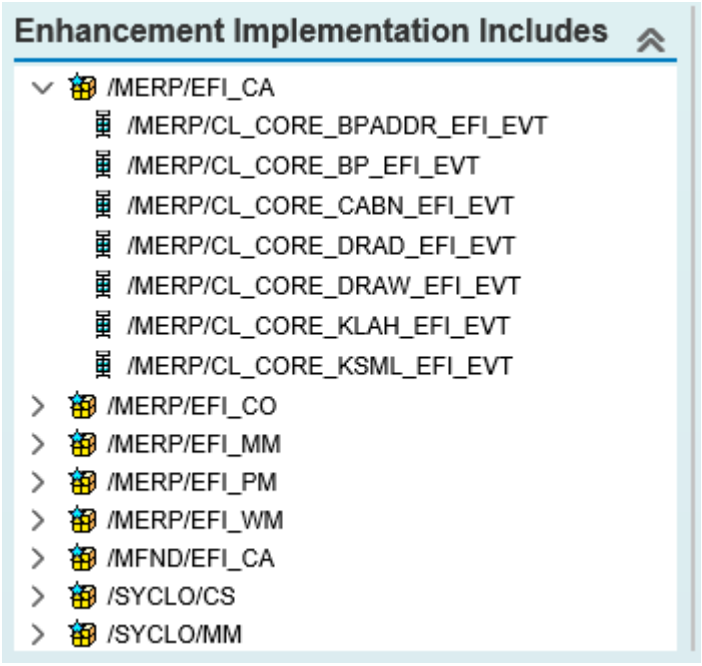
Enhancement Framework Implementation (EFI) source code plug-ins are implemented by the SAP Mobile Add-On for each business object where you configure change detection.

The source code plug-in is provided as an ABAP include file. Each exchange object is assigned to a plug-in to handle the actual change detection process. EFIs are typically available across multiple mobile applications running on the same system.

EFIs collect before and after images of data in an SAP object that was created, modified, or deleted. The EFI then hands those images to the exchange object, which continues with the data processing. Therefore, link the EFIs to their corresponding exchange objects.

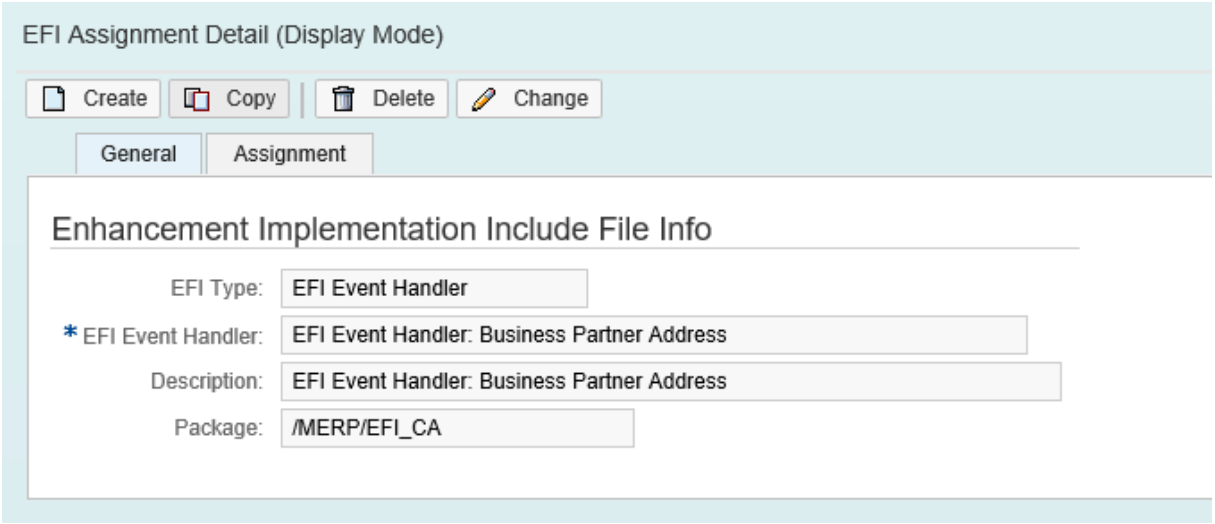
Enhancement Implementation Includes Section

The Enhancement Implementation Includes section is a tree of the include file list in the package. To expand the list, click the arrow to the right of the first item.



General Tab

Use the *General* tab to view and modify the general settings for chosen EFI file.

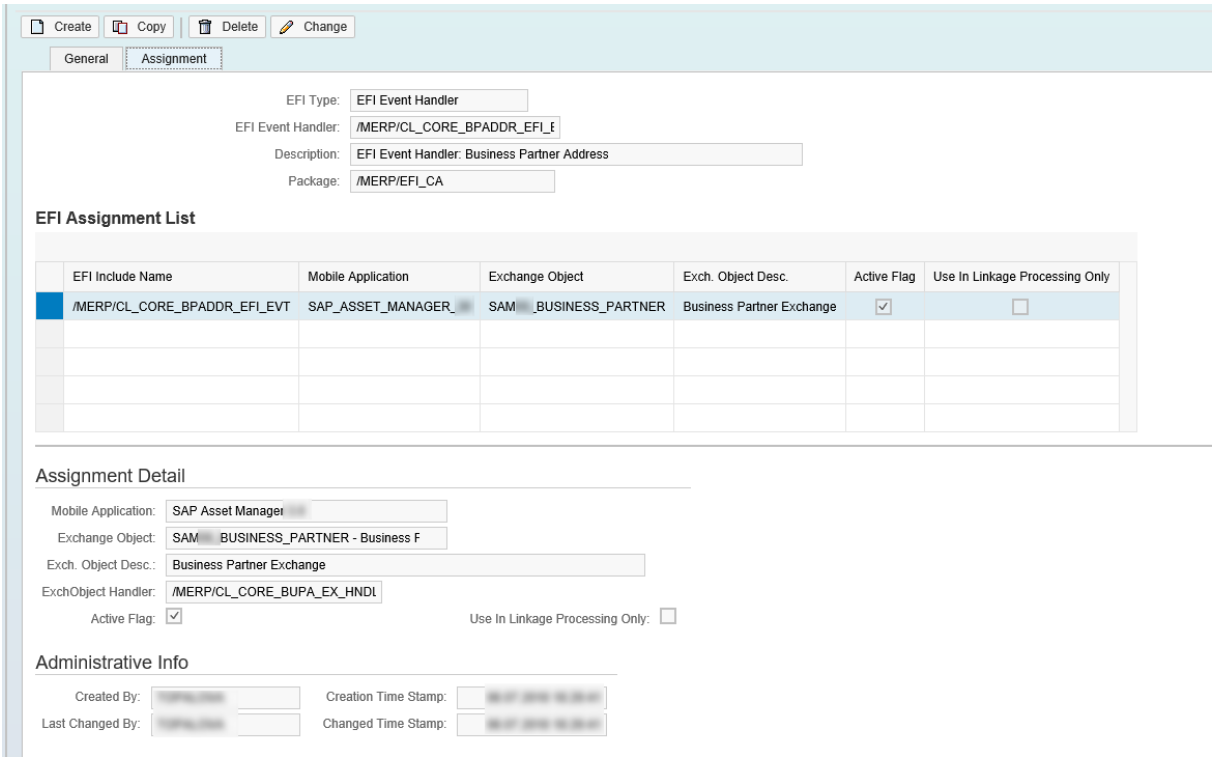


- **EFI Type:** Select one of two options; *Standard EFI Include* or *EFI Event Handler*. Choosing Standard EFI Include is the traditional way to implement EFI and configure the EFI assignments. Selecting EFI Event Handler implements EFI using an ABAP class-based approach.
  When you use a class-based approach, EFI implementation is developed as a subclass of `/SMFND/CL_CORE_EFI_EVENT_BASE`. Available EFI event handler classes are displayed in the dropdown field. The EFI class-based approach provides a more robust functionality and is recommended for a new EFI implementation.

- **EFI Include Name:** File name of the source code plug-in
- **Description:** Short description of the EFI. The description field is automatically populated when you select the EFI include name and is read only.
- **Package:** Package where the EFI is located. The package field is automatically populated when you select the EFI include name and is read only.

Assignment Tab

Use the *Assignment* tab to modify the EFI assignments.



- **EFI Information fields:** The EFI information fields at the top of the *Assignment* tab, like `<EFI Type>` and `<EFI Event Handler>`, are taken from information in the *General* tab and are read only.
- **EFI Assignment List:** Table that displays the plug-ins that are assigned to a specific include file. All column information is replicated in the *Assignment Detail* section directly below the table.
- **Mobile Application:** Read-only name of the specific mobile application
- **Exchange Object:** Name of the exchange object to which the EFI include file is assigned
- **Exchange Object Description:** Read-only description of the exchange object
- **Exchange Object Handler:** Read-only name of the class handler from the repository responsible for updating the exchange table
- **Active Flag:** When checked, the exchange object is in an active state. If unchecked, the EFI isn't linked to the assigned OMDO.
- **Use in Linkage Processing Only:** When checked, the xChange object is only allowed during linkage processing. If not checked, the original EFI is triggered during xChange processing.

2.2.5.2 Exchange Object Configuration

The exchange object defines what in the exchange table is updated in the exchange persistent layer, what class handler is called to update the exchange table, and what fields are related to the change detection.

Use the Configuration Panel to specify which changes are relevant to the mobile application and what conditions to satisfy for so that an update action is triggered. The *Exchange Object Configuration* panel has the following tabs:

- Technical Settings
- Change Detection Field Selection
- Change Detection Condition Filter
- Data Segment Settings
- Linkage Settings
- Push Settings

Technical Settings Tab

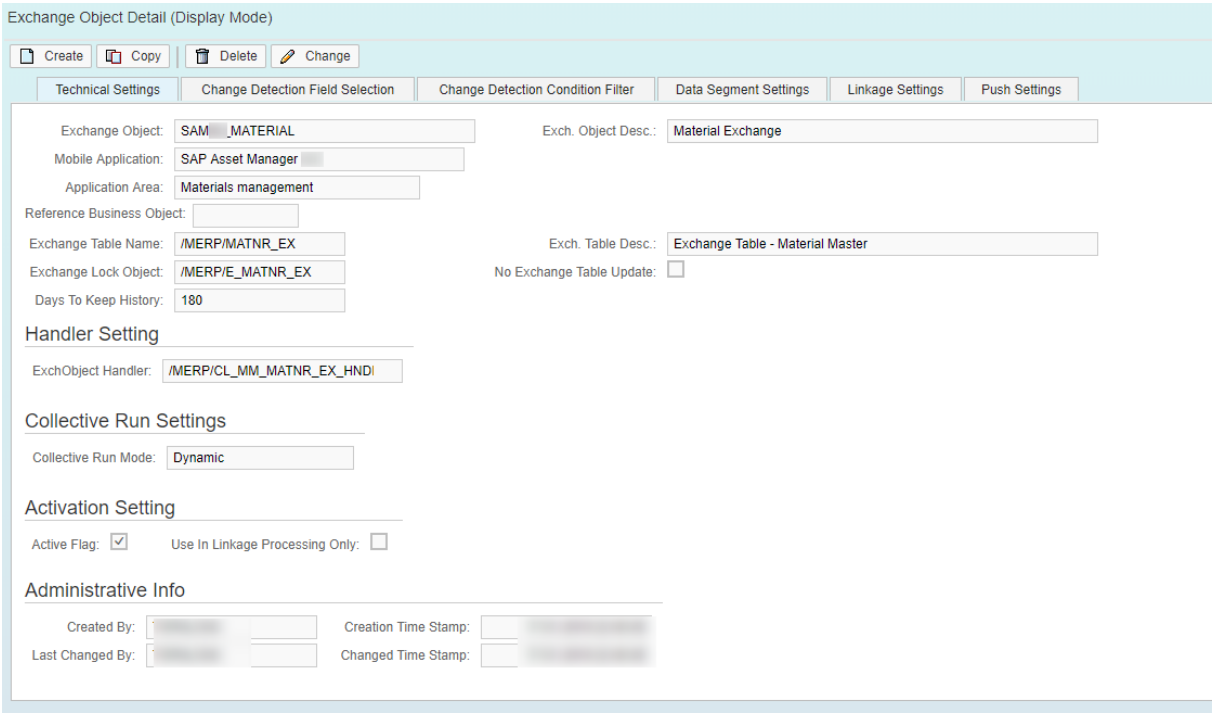
Use the Technical Settings tab to configure basic settings for an exchange object.

Use the `<Exchange Object>` field for the ID of the exchange object, limited to 40 characters. Type in a description in the `<Exchange Object Description>` field, limited to 60 characters. The `<Mobile Application>` field contains a dropdown where you can select your mobile application. The `<Application Area>` classifies the exchange object based on standard SAP application areas using a dropdown selection field.

The `<Reference Business Object>` is the standard SAP business object. The `<Exchange Table Name>` is the name of the table stored in SAP that contains the technical data. The `<Exchange Table Description>` is a brief description of the exchange table. The `<Exchange Lock Object>` field is used when updating the exchange table. Type in how many days you want to keep historical data in the `<Days to Keep History>` field. Check the `<No Exchange Table Update>` checkbox to not write the record to the exchange table in SAP when the record is changed.

- **Handler Setting:** Type in the name of the class handler from the repository that is responsible for updating the exchange table in the `<Exchange Object Handler>` field.
- **Collective Run Settings:** Define the condition where xChange processing is executed asynchronously as a V3 run by selecting one of the following mode options:
  - Dynamic: The collective run mode is determined at runtime by the xChange handler method `DETERMINE_EXEC_MODE`
  - Not Allowed: Not allowed to switch to collective run mode
  - Activated: Always execute asynchronously in V3 collective run mode
  - By User Parameter ID: Switch to V3 collective run mode for runtime user with the specified user parameter value set in the user profile
- **Activation Setting:** Check the `<Active Flag>` checkbox to ensure that the exchange object is in an active state. If unchecked, the exchange object performs no actions. When the `<Use in Linkage Processing Only>` checkbox is checked, the xChange object is only allowed during linkage processing and not if the original EFI was triggered during the xChange process.

The following screenshot shows an exchange process enabled for *MATERIAL*. Any changes for the MATERIAL master data are recorded in the exchange table and are transmitted to the client during the next transmit.



Change Detection Field Selection Tab

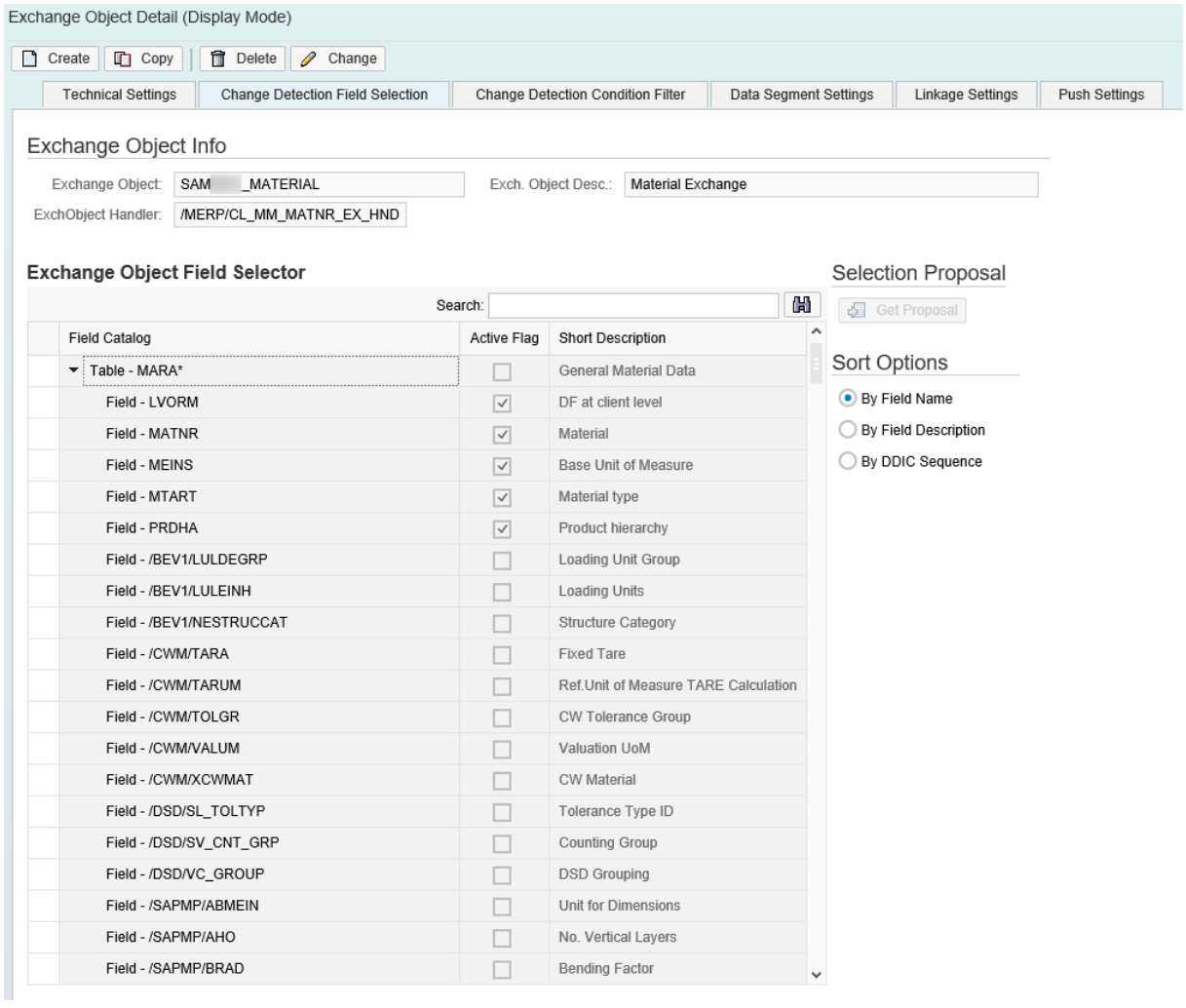
The Change Detection Field Selection tab lets you optimize the change detection process for a mobile application. If a value change is detected for any fields within the group, the object identifier is written to the exchange table, indicating that a change was made. If the `<Active Flag>` is not checked for a field, any value changes made to that field are not detected and recorded to SAP during the exchange process. By default, all fields are initially checked.

The *Exchange Object by Application* tree lists all application areas and the exchange objects linked to each application area. Expand the tree by clicking on the arrows to the right of the application area to display the exchange objects associated with it.

- **Exchange Object Info:** The `<Exchange Object>` field is read only and is the ID of the exchange object. The `<Exchange Object Description>` is read only and is a brief description of the exchange object. The `<Exchange Object Handler>` field is read only and is the name of the class handler from the repository that is responsible for updating the exchange table.
- **Exchange Object Field Selector:** The `<Field Catalog>` column is comprised of non-editable rows of all fields that are detected by the class handler when changes are made. These fields are grouped by the technical table name of the SAP business object.
  When the `<Active Flag>` checkbox is checked, either the table or a field within the table is active. Any value change to the selected field is detected by the class handler. Note that if you check the Active Flag checkbox on a table row, it selects all the rows within the table.
  The `<Short Description>` is a read only field that contains a brief description of the table or of a field withint the table.

- **Selection Proposal:** In a typical mobile application installation, you do not want to have all fields marked as active for change detection. Rather, only the fields that are active on the odata mobile data object that are brought down to the mobile device will also be active in the exchange object. Based on odata mobile data object usage in the application, the selection proposal examines the active flags that are checked for an exchange object's table fields and provides recommendations to the administrator on which fields should be checked or unchecked.

See the following screenshot for an example of the enabled exchange object *MATERIAL*, where the properties of the object are captured and recorded in the exchange table. The properties that trigger the exchange are defined on this *Change Detection Condition Filter* tab, as seen in the checked `<Active Flags>`:

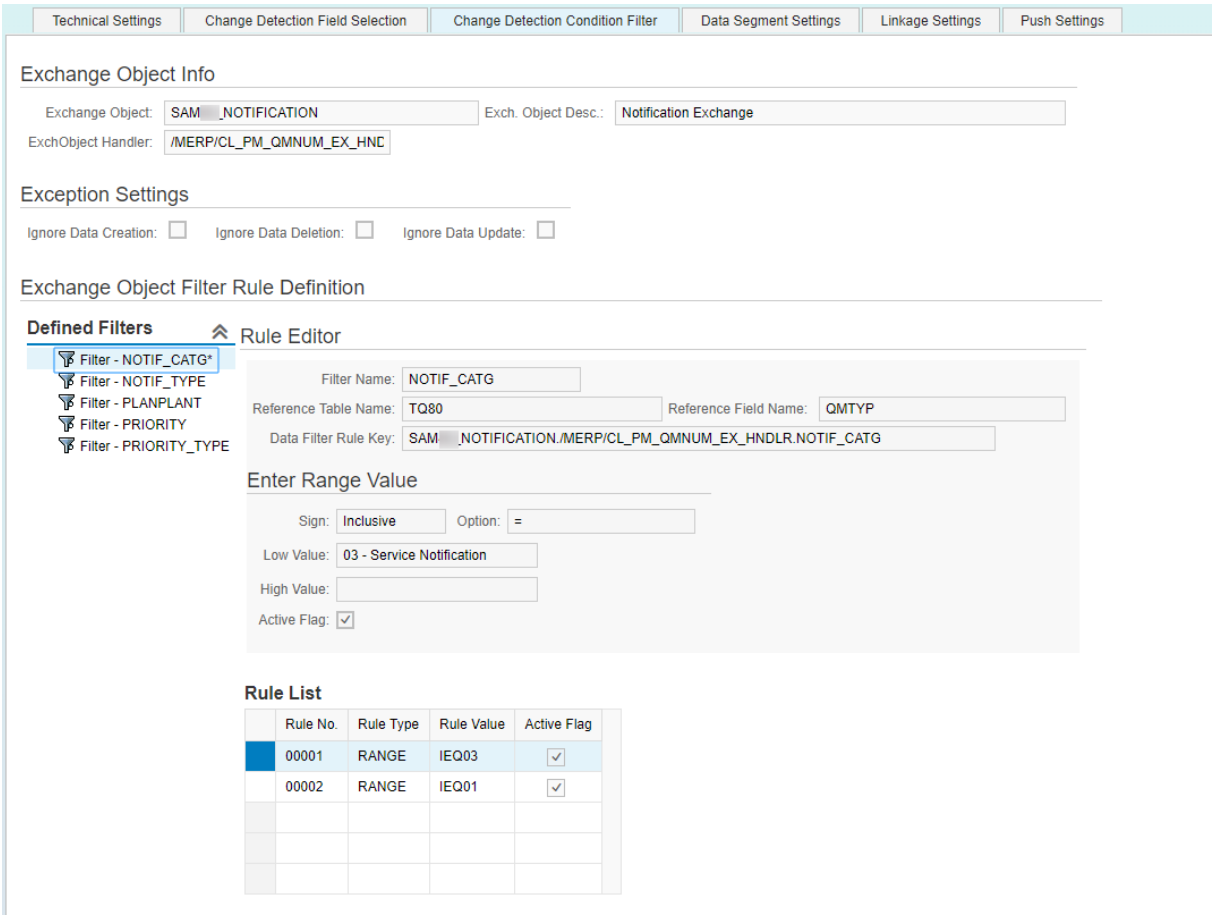


Change Detection Condition Filter Tab

The Change Detection Condition Filter tab lets you restrict change detection based on data content. For exchange handlers to support this feature, define data filter conditions for which the underlying SAP business object must qualify before the change detection process is triggered. The condition is defined at the table field level and is in the SAP range table format.

- **Exchange Object Info:** The `<Exchange Object>` field is read only and is the ID of the exchange object. The `<Exchange Object Description>` is read only and is a brief description of the exchange object. The `<Exchange Object Handler>` field is read only and is the name of the class handler from the repository that is responsible for updating the exchange table.
- **Exception Settings:** When the `<Ignore Data Creation>` checkbox is checked, newly created records and data are not processed to the exchange table. When the `<Ignore Data Deletion>` checkbox is checked, deleted records and data are not processed to the exchange table. When the `<Ignore Data Update>` checkbox is checked, updated records and data are not processed to the exchange table.
- **Defined Filters:** Lists all the data filters supported by the class handlers.
- **Rule Editor:** The `<Filter Name>` is read only and is the name of the filter as defined by the class handler developer in the class handler method. The `<Reference Table Name>` is read only and is the technical name of the SAP database table field where the filter is applied as defined by the class handler developer. The `<Reference Field Name>` is read only and is the technical name of the SAP database table field where the filter is applied as defined by the class handler developer. The `<Data Filter Rule Key>` is an internal technical key used by the framework at runtime.
  Use the values in the *Enter Range Value* section to set the range. The `<Sign>` field is the value for the SAP range table column SIGN. The `<Option>` field is the value for the SAP range table column OPTION. The `<Low Value>` field is the value for the SAP range table column LOW. The `<High Value>` field is the value for the SAP range table column HIGH. When the `<Active Flag>` checkbox is checked, the rule is active. The *Rule List* is a table that displays a list of rules that are defined using the values in the *Rule Editor* and the *Range Value* sections.

The following screen shows that any exchange detected for the exchange object *NOTIFICATION* will be considered only if the notification is maintained in one of the roles defined in the *NOTIF_CATG* criteria.



2.2.6 Push Framework Settings

2.2.6.1 Push Scenario Definition

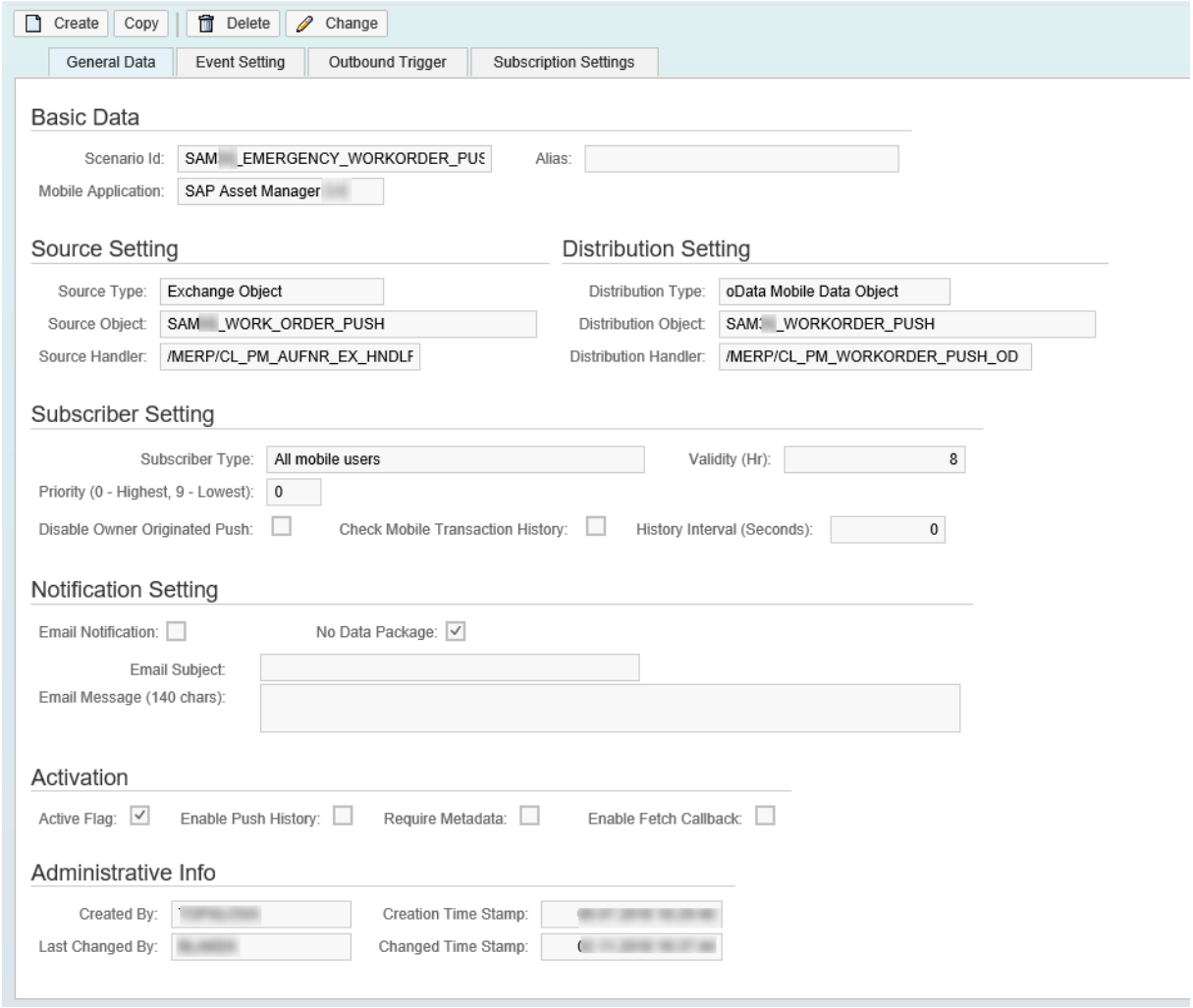
Push scenarios define the trigger conditions, type of data, the mobile users receiving the data, and the users for the data.

A mobile client typically synchronizes with the SAP system by initiating a synchronization request to download the latest application data from the SAP system. Some mobile applications require the SAP system to send application data or push notifications to the client when certain trigger conditions are met. If these trigger conditions are not present, the mobile client does not initiate the synchronization request.

You define trigger conditions through the creation of push scenario definitions. Use the tabs found in the *Push Scenario Definition* page to configure a push scenario. The *Push Scenario Definition* page contains the following tabs:

- General Data
- Event Setting
- Outbound Trigger
- Subscription Settings

General Data Tab



You can define the following attributes in the *General Data* tab:

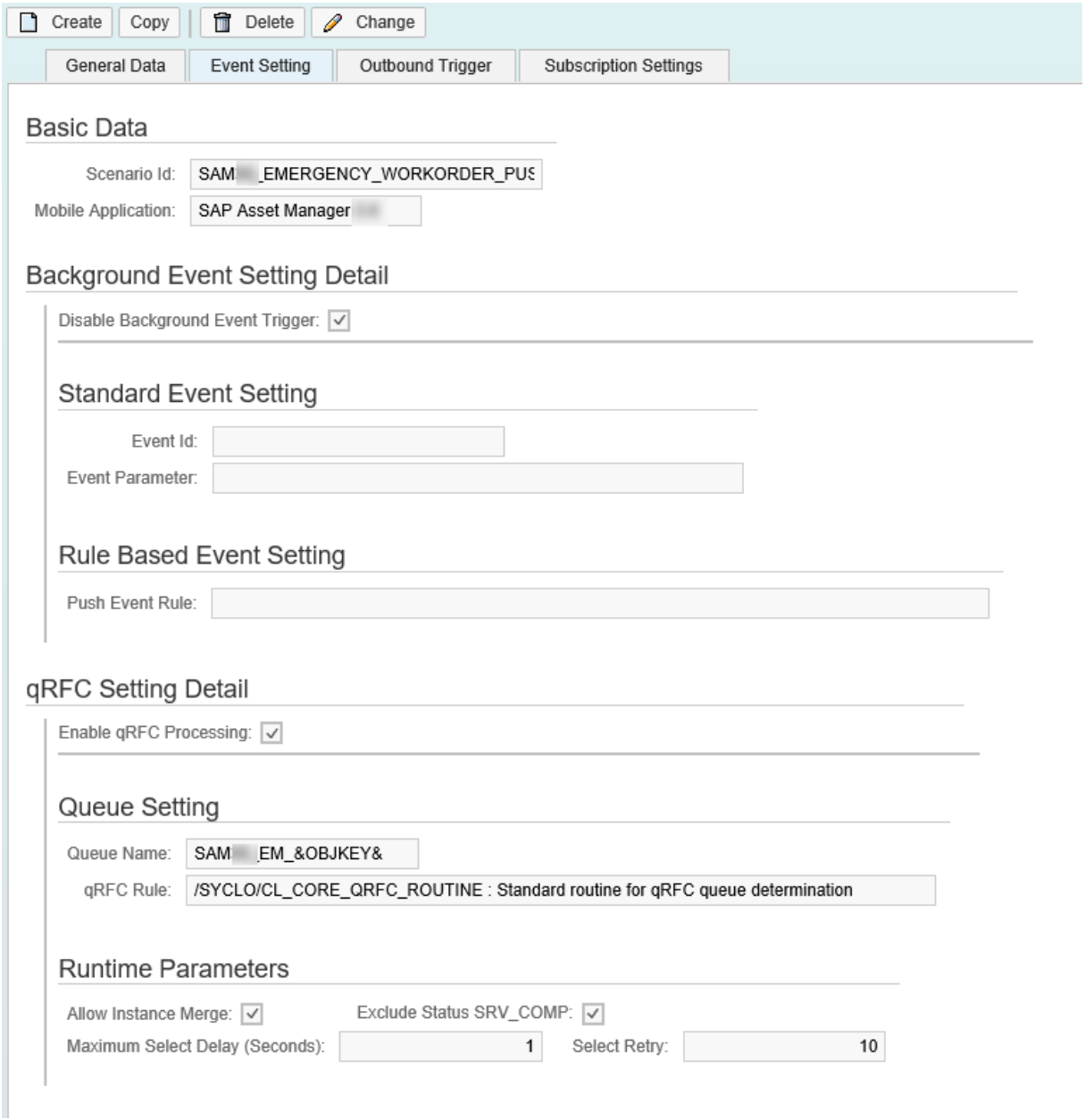
- **Basic Data section:** Enter the ID of the push scenario in the required `<Scenario ID>` field, which is limited to 40 characters with namespace protection. Use either a *Y* or a *Z* namespace. Ensure that the ID is unique in the SAP system. Enter the name of the mobile application in the `<Mobile Application>` field, limited to 40 characters. Give an optional `<Alias>` to the push scenario. Multiple push scenarios can share the same alias, to allow central processing on the client side.
- **Source Setting section:** The `<Source Type>` defines how to trigger the push scenario. Two options are supported:
  - xChange Object: The push scenario is triggered when qualifying data is changed in the SAP system and change conditions defined in the xChange object are detected.
  - Client on Demand Request: The push scenario is triggered based on a request from the mobile client. No data change in the SAP system is required. The client on demand request is not available for OData based mobile applications.

  The `<Source Object>` applies to the source type of the xChange object. The xChange object determines the data change trigger for the push scenario. The `<Source Handler>` is the xChange handler assigned to your selected xChange object.

- **Distribution Setting section:** The `<Distribution Type>` defines what application data is sent and how recipients are determined. Two options are supported:
  - Mobile Data Object: Not supported for the SAP Asset Manager application
  - OData Mobile Data Object: Use *OData Mobile Data Object* for the SAP Asset Manager application. The assigned OMDO determines the recipients for the push notification.

  When you select *OData Mobile Data Object* as a Distribution Type, assign an OMDO configuration object to the OMDO in the `<Distribution Object>` field. The `<Distribution Handler>` is the OMDO handler associated with the assigned OMDO configuration object.
- **Subscriber Setting section:** The `<Subscriber Type>` determines the user pool. The user pool is used to determine who receives the push notification. Enter the amount of time the push instance generated for the push scenario is valid, in hours, in the `<Validity>` field. The `<Priority>` is the processing priority of the push instance. A push instance with a higher priority is processed first.
  When the *Disable Owner Originated Push* checkbox is checked, the user who made the change to the SAP data is excluded from the push recipient list triggered by the data change. When the *Check Mobile Transaction History* checkbox is checked, the SAP system only sends a push to the user if the user has previously downloaded the same SAP object and the object is still valid. This setting is commonly used for pushing time sensitive data such as workflow tasks to users.
  Enter the `<History Interval>`, in seconds, to determine whether the object received by the user through a previous push is still valid. No push is sent to a user who has expired push data. The *History Interval* setting is linked to the *Check Mobile Transaction History* setting.
- **Notification Setting section:** When the *Email Notification* checkbox is checked, an email notification is generated to push recipients during push processing. The email address used is based on information defined in SAP user master or mobile user master data defined in the Administration & Monitoring Portal. If email notification is enabled, type a subject in the `<Email Subject>` field. If email notification is enabled, type your e-mail in the `<Email Message>` field.
  When the *No Data Package* checkbox is checked, no data payload is generated by push processing. Check this box to generate only push notifications. OData-based mobile applications such as SAP Asset Manager only support push notifications without data payloads. Traditional, Agentry-based, mobile applications can support a data payload during push processing.
- **Activation section:** If the *Active Flag* checkbox is checked, the push is active. If the *Enable Push History* checkbox is checked, information about push recipients and data object keys are recorded in the push history table. If the *Require Metadata* checkbox is checked, technical data for the push data payload is generated. If the *Enable Fetch Callback* checkbox is checked, the mobile client retrieves data about the pushed SAP object using a regular sync request instead of precalculating the push data payload during push processing. Enabling push callback can ensure the client always receives up-to-date SAP data when receiving a push.

Event Setting Tab

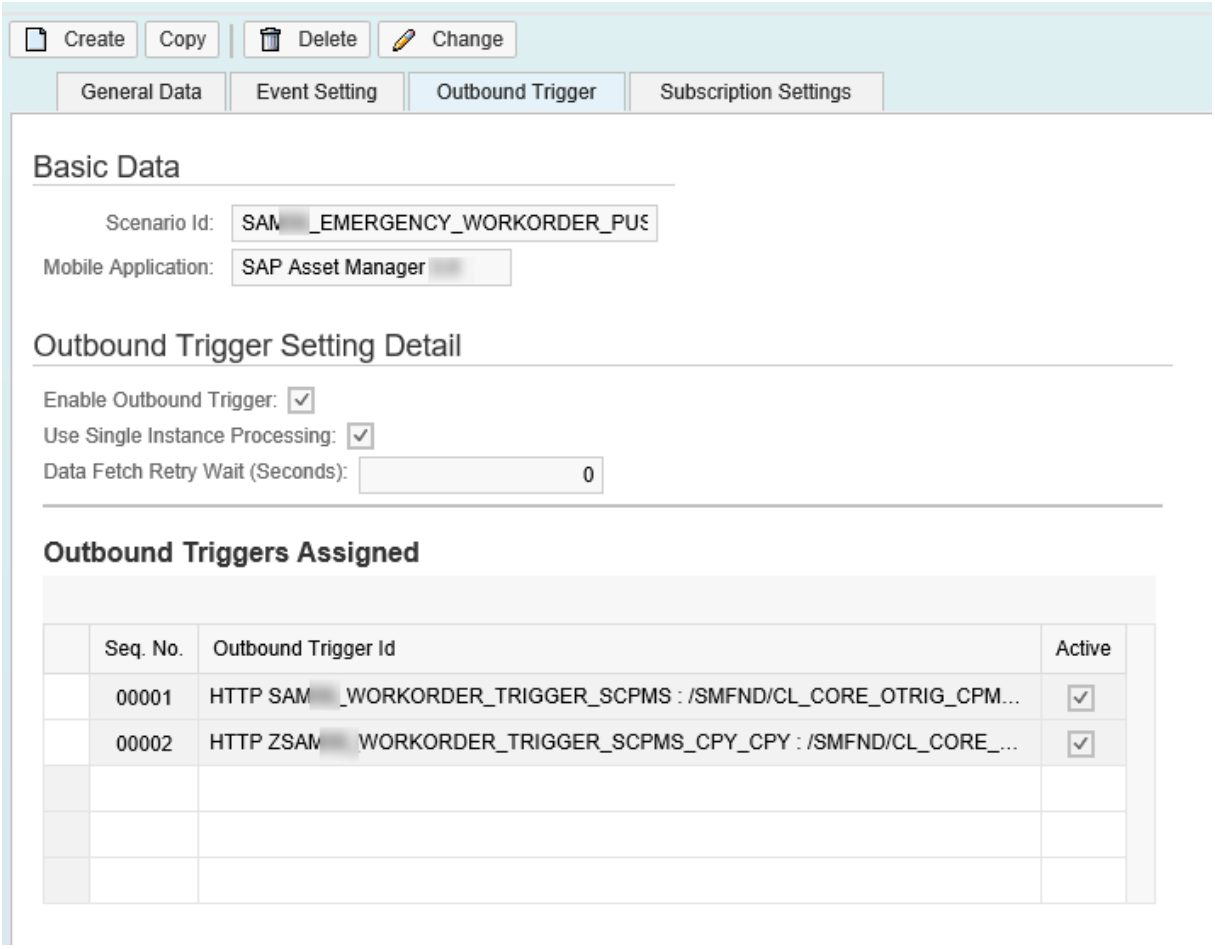


You can define the following attributes in the *Event Setting* tab:

- **Background Event Setting Detail setting:** If the *Disable Background Event Trigger* checkbox is *not* checked, a background event is raised during push processing.
- **Standard Event Setting:** The `<Event ID>` is the background event ID that is raised. The `<Event Parameter>` is the background event parameter.
- **Rule Based Event Setting:** The `<Push Event Rule>` is a routine that generates a dynamically formatted event ID and parameter based on supported runtime variables.
- **qRFC Setting Detail:** If the *Enable qRFC Processing* checkbox is checked, push processing is handled in the background as a qRFC call.

- **Queue Setting:** Enter the qRFC queue name used for push processing in the `<Queue Name>` field. The `<qRFC Rule>` is the routine that can generate a dynamically formatted qRFC queue name based on supported runtime variables.
- **Runtime Parameters:** If the *Allow Instance Merge* checkbox is checked, the creation of a new push instance cancels existing push instances for the same SAP object key that are not completed. If the *Exclude Status SRV_COMP:* checkbox is checked, push instances with a status of SRV_COMP are not reprocessed. If it is not checked, push instances with a status of SRV_COMP are reprocessed.
  Type in the time delay before processing a push sequence in the `<Maximum Select Delay>` field. The maximum select delay is used in qRFC push processing. Use this setting if there could be a time delay in the push instance registry table database update. The delay could allow the push registry data to fully update before the push processing starts.
  Type in the number of times the push processor should try to read push instance information from the push registry if the read failed into the `<Select Retry>` field.

Outbound Trigger Tab

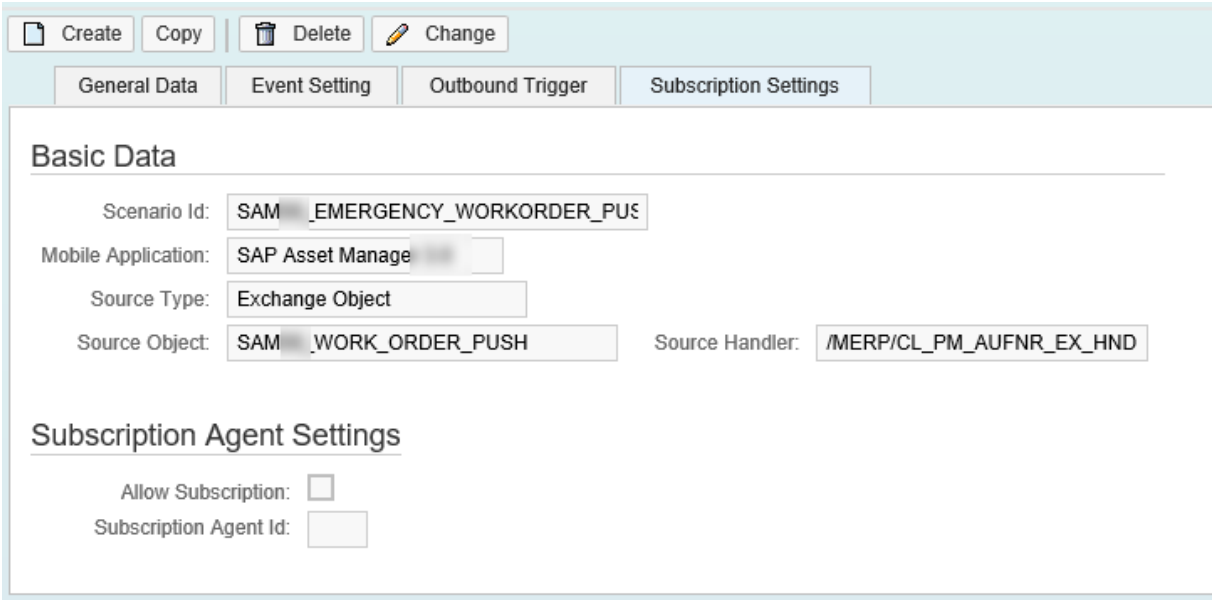


Outbound triggers handle interfacing with external systems. You can assign multiple outbound triggers to a push scenario. Assigned outbound triggers are invoked at the end of push processing, based on the assigned sequence.

Check the *Enable Outbound Trigger* checkbox to activate the selected outbound trigger.

Subscription Settings Tab

To allow an on-demand subscription based push request from the mobile client, define the subscription setting in the *Subscription Settings* tab. Subscriptions allow the mobile client to trigger a push process instead of a traditional trigger by the back end SAP system update. OData based mobile applications do not support subscription-based on-demand push configuration.



- **Allow Subscription:** Check to enable subscription-based push processing for the push scenario
- **Subscription Agent ID:** Displays the subscription agent assigned to handle the subscription request

2.2.6.2 Outbound Trigger Configuration

Outbound triggers allow a mobile application to interface with external systems such as the SAP Cloud Platform.

You can integrate outbound triggers into one of the following mobile application processes:

- Push processing
- OData mobile data object processing

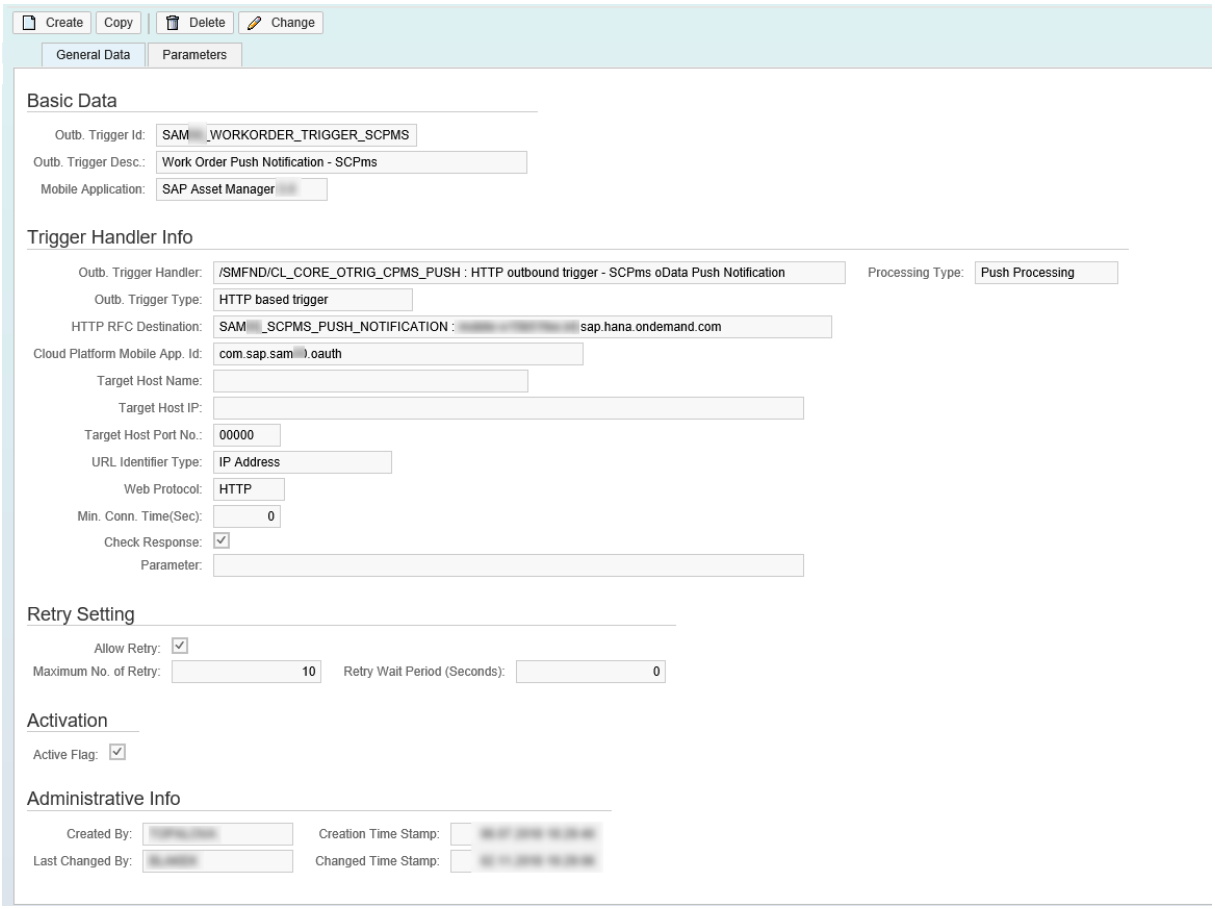
An outbound trigger can support only one of the two available processes. The process is determined by the outbound trigger handler. An outbound trigger handler can support any of the interface technologies, such as HTTP triggers, file triggers, and web service triggers.

The *Outbound Trigger* page contains the following tabs:

- General Data

- Parameters

General Data Tab



You can define the following attributes in the *General Data* tab:

Basic Data

- **Outbound Trigger ID:** Required field. Unique ID of the outbound trigger in the Y or Z namespace, limited to 40 characters.
- **Outbound Trigger Description:** Short description of the outbound trigger
- **Mobile Application:** Select your mobile application. The outbound trigger configuration detail is defined for the individual mobile application.

Trigger Handler Info

- **Outbound Trigger Handler:** ABAP OO class that provides the technical implementation for the outbound trigger. The outbound trigger handler must be a subclass of `/SYCLO/CL_CORE_XXX` You can reuse an outbound trigger handler to provide technical implementation for multiple outbound triggers.
- **Processing Type:**
- **Outbound Trigger Type:** Selection depends on the implementation of the outbound trigger handler

- **HTTP RFC Destination:** Only used when *HTTP-based trigger* is selected as the outbound trigger type. The HTTP RFC destination establishes a connection for the outbound trigger
- **Cloud Platform Mobile App ID:** used to interface with SAP Cloud Platform mobile services. The ID identifies the mobile application ID that was set up in SCPms.
- **Target Host Name:** Host name the outbound trigger is connecting to
- **Target Host IP:** IP address of the host to which the outbound trigger is connecting to
- **Target Host Port Number:** Port number of the host to which the outbound trigger is connecting to
- **URL Identifier Type:** Defines the information to use by the outbound trigger to make the connection
- **Web Protocol:** HTTP or HTTPS
- **Minimum Connection Time (Seconds):** Wait time before checking the response to an outbound trigger request from the remote system
- **Check Response:** If checked, a check response to an outbound trigger HTTP request is active
- **Parameter:** General purpose parameter that provides input information to the outbound trigger handler
- The following fields are only visible after other fields are selected:
  - Logical File Name: Used by outbound trigger type *File Based Trigger*
  - RFC Destination: Used by outbound trigger type *Remote Function Call*
  - RFC User ID: Used by outbound trigger type *RFC User ID*. User ID to make the RFC call.

Retry Setting

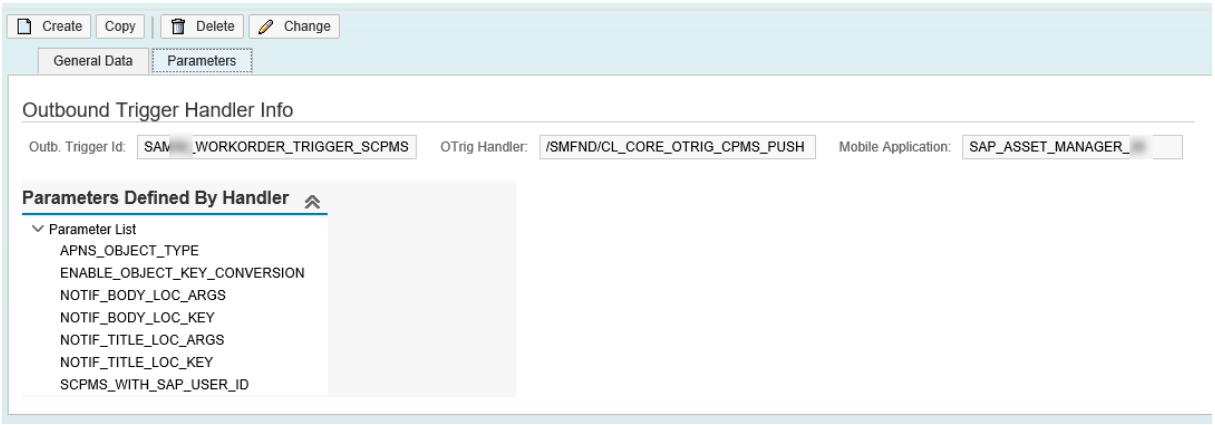
- **Allow Retry:** If checked, the outbound trigger is allowed to rerun
- **Maximum Number of Retry:** Set the maximum number of times the outbound trigger can rerun
- **Retry Wait Period (Seconds):** Set the minimum wait time between output trigger retries

Activation

If the *Active Flag* checkbox is not checked, the outbound trigger is not enabled.

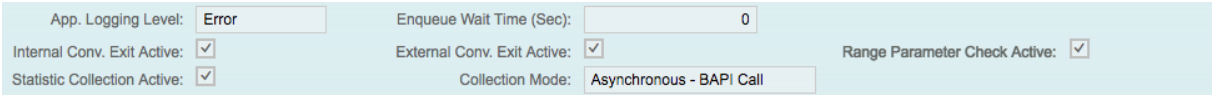
Parameters Tab

An outbound trigger handler can declare special purpose parameters. If parameters are declared, they are displayed in the *Parameters* tab. You can declare any number of parameters. A parameter can be a single field parameter or a structured record.



2.2.7 Technical Settings

Technical settings affect all components of the framework.

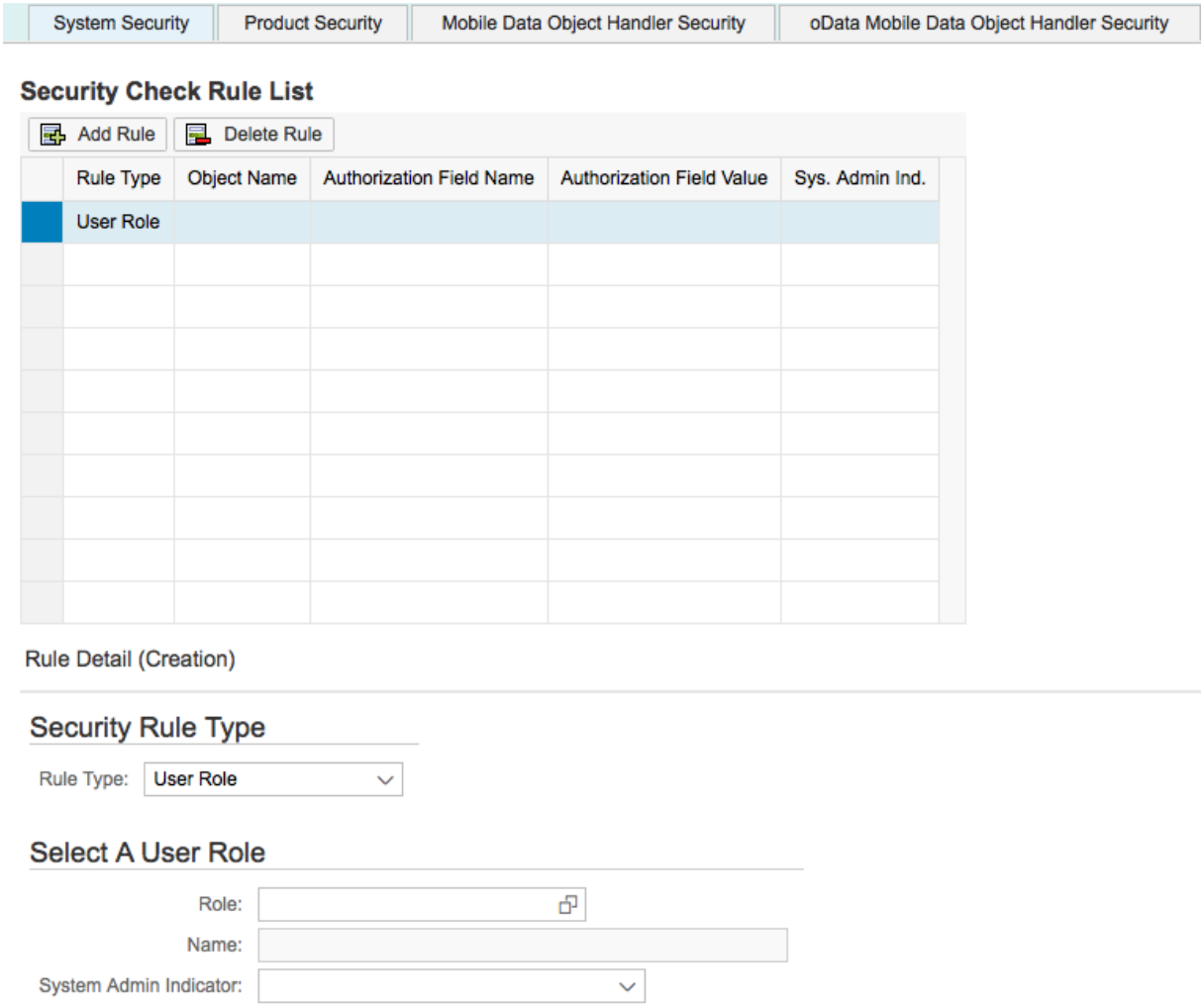


The following fields make up the Technical Settings page:

- **Application Logging Level:** Defines the logging level for all framework components. Logging entries are recorded in the SAP application log database under the object `/syclo/`. The logging levels are:
  - No logging
  - Abort
  - Error
  - Warning
  - Info
  - Debug
  - Trace
- **Enqueue Wait Time (Sec):** The Enqueue Wait Time parameter controls the number of seconds the underlying component should continue to try to access a locked SAP object in intervals of 1 second during an update by a mobile device. The update process aborts if accessing the locked object is still unsuccessful after the wait time.
- **Internal Conversion Exit Active:** When checked, the framework runtime data manager performs a standard SAP external-to-internal format conversion exit for all inbound BAPI parameters. The option is enabled by default. An application developer should only change this setting as it has a direct impact to the SAP Asset Manager application.
- **External Conversion Exit Active:** When enabled, the framework runtime data manager performs standard SAP internal-to-external format conversion exit for all outbound BAPI parameters. This option is enabled by default. An application developer should only change this setting as it has a direct impact to the SAP Asset Manager application.
- **Range Parameter Check Active:** When enabled, the framework runtime data manager performs checks on all SAP range parameters of inbound BAPI parameters. The SAP range parameter has the structure of SIGN, OPTION, LOW and HIGH. If SIGN and OPTION are not specified, a check routine sets SIGN to I and OTPION to EQ. This option is enabled by default. An application developer should only change this setting as it has a direct impact to the SAP Asset Manager application.
- **Collection Mode:** Collection mode determines how system statistic records are written to the database. Two modes are supported currently: Synchronously and Asynchronously. When you select Synchronously, the statistics record is written to the database in real-time during BAPI calls. However, selecting this option incurs a performance penalty. Selecting Asynchronously means that statistics are collected in-memory and written asynchronously to the database at the end of the BAPI call.
- **Statistic Collection Active:** When enabled, the framework records all runtime statistics associated with the BAPI calls between the middleware server and SAP. This collection provides data for the KPI statistics collections found in the Administration portal. An application developer should only change this setting as it has a direct impact to the SAP Asset Manager application.
- **Created By, Creation Time Stamp, Last Changed By, Changed Time Stamp:** The user ID and time stamps are automatically logged when a record is created or changed.

2.2.8 Mobile Authorization Settings

You can define security rule settings for the Mobile Integration Framework for SAP and mobile applications as well.



All security checks are carried out by the Mobile Integration Framework at runtime, with checks performed at the following levels:

- *System*
  Application independent. Applies to all components built on the Mobile Integration Framework.
- *Product*
  Security at the mobile application and product level
- *Mobile Data Object Handler*
  Specific to a Mobile Data Object class handler
- *OData Mobile Data Object Handler*
  Specific to an OData Mobile Data Object class handler

The following types of security rules can be defined:

- *User Role*
  Rules based on predefined user roles

- *Authorization Profile*
  Rules based on predefined authorization profiles.
- *Authorization Object*
  Rules based on predefined authorization objects.

Special Security Role with System Indicator

You can define special security rules using user roles. These security rules can be assigned with system indicators. These special security rules with system indicators are used to limit access to the ConfigPanel and Administration & Monitoring tools. The following system indicators are available:

- *System Administrator*
  If security rules are defined, only users with the required user role can have full access to the Administration & Monitoring tool.
- *System Administration – View Only*
  If security rules are defined, only users with the required user role can have read access to the Administration & Monitoring tool.
- *System Configurator*
  If security rules are defined, only users with the required user role can have full access to the ConfigPanel.
- *System Configuration – View Only*
  If security rules are defined, only users with the required user role can have read access to the ConfigPanel.

3 Mobile Add-On Configuration Panel Common Procedures

3.1 Mobile Application Configuration Procedures

3.1.1 Mapping Work Orders to a STARTED Status - Overview

By default, the SAP Asset Manager application maps the *STARTED* work order status on the client to the *REL* status in SAP Mobile Add-On.

In many implementations, a status of *MOBI* is used in SAP Mobile Add-On to indicate that the work order is started by a technician. The *MOBI* status cannot be modified on the back end.

You can map the mobile status to a different status within SAP Mobile Add-On by altering the mobile application configuration for SAP Asset Manager and changing the system status technical code for the STARTED mobile status. After you change the system status technical code, updates to SAP Mobile Add-On made when a user starts a work order set the status in SAP Mobile Add-On to the MoBI status, matching the entered technical code.

The only modification to make is in the ConfigPanel, in the *Mobile Application Configuration* page, *Mobile Status Setting* tab. Change the mobile status for a started work order in the list of the mobile status options for SAP Asset Manager, with the system status value of that same record altered to use the technical code of the desired status.

3.1.1.1 Changing the Mapping of a Mobile Status to STARTED

Prerequisites

Address the following before performing this procedure:

- Determine and note the technical code of the work order system status to which the mobile status STARTED will be mapped, as it is used in the procedure.
- The system status to which you are mapping the mobile status of STARTED in this procedure is configured as a work order status.
- The person performing this procedure has access to the ConfigPanel and permissions to change configuration settings of the elements within it.

Context

The following procedure describes the steps required to change a system status when a mobile STARTED status is mapped to it.

Procedure

1. Starting from the ConfigPanel home page, click the *Mobile Application Configuration* link. Then click the *Mobile Status Setting* tab.
2. Choose your desired mobile application from the list of *Defined Mobile Applications* in the left pane.

   The application level status settings display in the tab to the right. Information includes the *Mobile Status List*.
3. In the *Mobile Status List* table, find the *Object Type* of `<WORKORDER>` with a *Mobile Status* of `<STARTED>` and click the *Change* button.
4. Change the *System Status* value to the technical code of the system status to which the STARTED mobile status should be mapped. When done, click *Save*.

Results

After completion of the procedure, the STARTED mobile work order status is mapped to a different system status than the default REL status.

3.1.2 Configuring CATS and PM Confirmation Time Entries

Use parameters to configure CATS and PM confirmation minute interval values.

Context

Use the *CatsMinuteInterval* parameter when CATS is enabled and the *LaborTimeMinutesInterval* parameter when PM confirmations are enabled. The procedure below is the same for either parameter, even though this guide is using the *CATSMinuteInterval* parameter as an example.

When a mobile user manually logs their time, or their time is automatically logged for them through the use of the application, the time logged is rounded to the nearest interval configured. For example, you manually log an additional 12 minutes of work on a work order on a mobile device. Your *CATSMinuteInterval* parameter is set to *15*. Therefore, your additional time logged is automatically rounded up to 15 minutes. The time entry screens will also have their duration control values limited to minute values matching the configured interval.

The default value for both parameters is *15*.

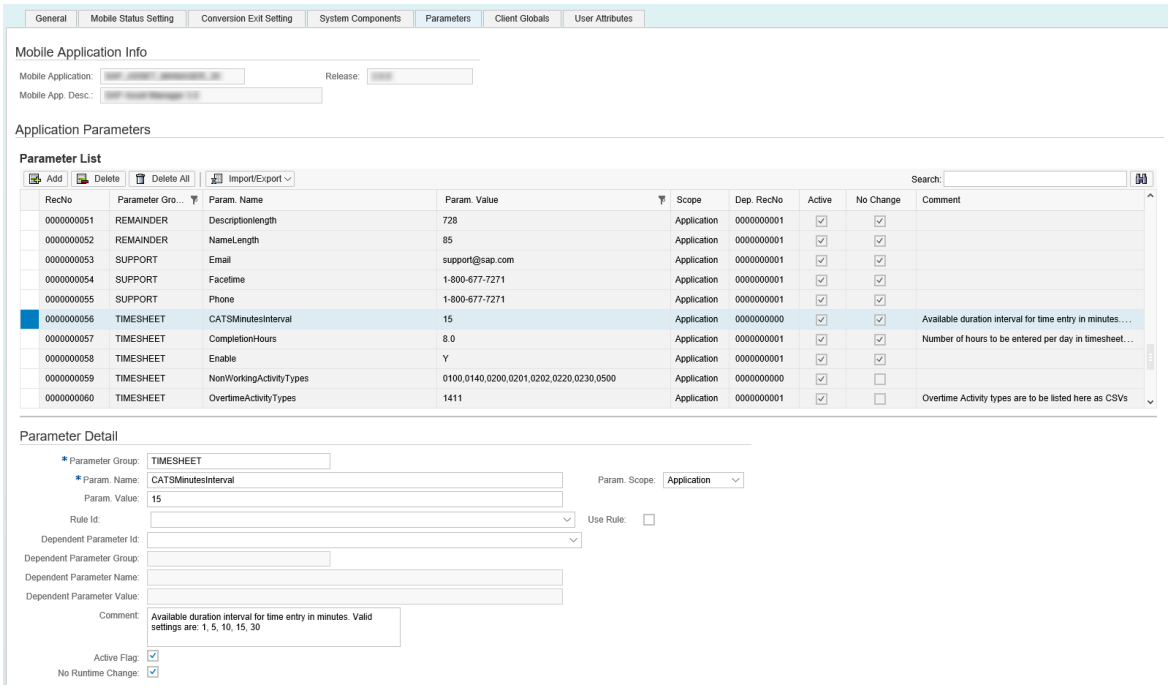
Procedure

1. Using the ConfigPanel, navigate to *Mobile Application Configuration* > *Parameters tab*. In the left column, *Defined Mobile Applications*, select your application.

   The *Parameter List* populates with a list of all parameters available for the application.

2. The *CATSMinuteInterval* parameter is found in the *TIMESHEET* group. You can scroll down to find the parameter, or perform a search using the *Search* box. Highlight the *CATSMinuteInterval* parameter and click the *Change* button.

   > **i Note**
   >
   > You can find *LaborTimeMinutesInterval* parameter in the *PMCONFIRMATION* group.



3. You can change either the CATS or the PM confirmation labor time interval to the following values:
   - 1
   - 5
   - 10
   - 15
   - 30

   If you accidentally set the parameter to an interval value that is not an allowed value, the parameter automatically defaults to a value of *15* on the client device.

4. Check the `<Active>` flag to ensure that the parameter is used by the mobile application. If desired, and if not already checked, check the `<No Runtime Change>` box to ensure that the value of the parameter is not overridden at runtime through synchronization processing.

5. *Save* your changes.

3.1.3 Configuring Clock In Clock Out

The Clock In Clock Out (CICO) feature decouples time tracking from the mobile status of a work order or operation, allowing multiple users to start and log time against the same work order or operation simultaneously.

Overview

The CICO parameter in the ConfigPanel allows multiple users to work on the same work order or operation, where all users receive the work orders and operations to their devices. CICO allows multiple work orders and operations to be in the Started state that belong to different users. Mobile device users are able to see the CICO state of all work orders or operations on their device. Users can also filter their work order or operations lists based on clock in or clock out status.

If CICO is enabled in the ConfigPanel:

- Multiple people can work on the same work order or operation even if the work order or operation is already started by another user
- Users can clock in to any work order or operation on their device
- Users can only clock in to one work order or operation on their device at a time
- Users must clock out of the current work order or operation before clocking in to a different work order or operation
- All time recording (CATS and Confirmation) uses the clock in clock out period as the default duration in time entry screens
- When a user clocks in to a work order or operation:
  - The timestamp of the work order or operation is saved to a user-specific table that is persisted in the back end
  - The mobile status of the work order or operation is set to Started if it isn't already in a started state
- When a user clocks out of a work order or operation:
  - The work order or operation status is set to either Hold or Complete
  - If a work order or operation is set to Complete and Confirmations are used, the user can set it as the final confirmation

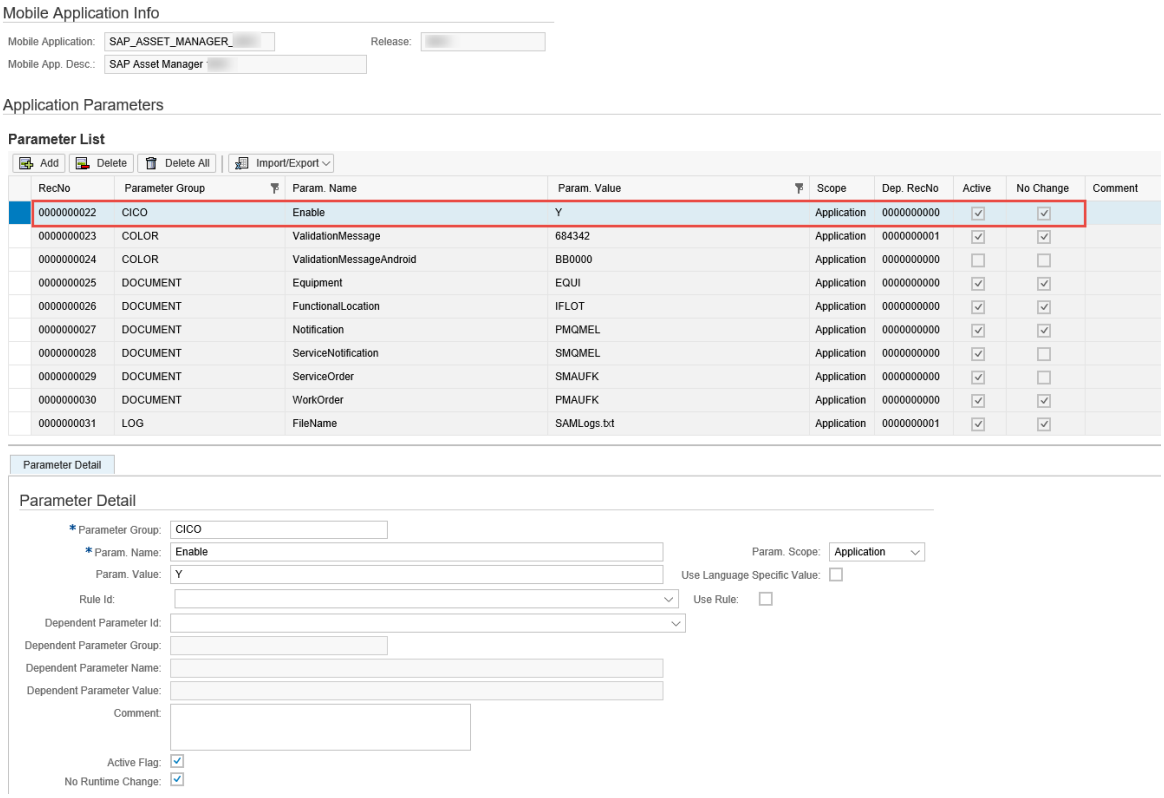
If CICO is disabled in the ConfigPanel:

- A user can start any work order or operation that is in a Hold, Received, or Local state
- A user can start only one work order or operation at a time
- The mobile status of a work order or operation is used to track time in either CATS or Confirmations

How to Set the CICO Parameter Using the Configuration Panel

1. Using the ConfigPanel, navigate to *Mobile Application Configuration* > *Parameters tab*. In the left column, *Defined Mobile Applications*, select your application.
   The *Parameter List* populates with a list of all parameters available for the application.

2. The *Enable* parameter is found in the *CICO* group. You can scroll down to find the parameter, or perform a search using the *Search* box. Highlight the *Enable* parameter and click the *Change* button.



3. If the *Enable* parameter is set to *Y*, clock in clock out is enabled. If it's set to *N*, the parameter is disabled.
4. Check the `<Active>` flag to ensure that the parameter is used by the mobile application. If desired, and if not already checked, check the `<No Runtime Change>` box to ensure that the value of the parameter isn't overridden at runtime through synchronization processing.
5. *Save* your changes.

3.1.4 Configuring Notification Catalog Types

Use parameters to configure the notification catalog types.

Context

Code groups that belong together in terms of content are grouped in catalogs. These catalogs are identified by the catalog type (a number or a letter). For example, in this way, you combine:

- All code groups for particular problems for a catalog type
- All code for causes for another catalog type and
- All code groups for activities for a further catalog type

Use the *CATALOGTYPE* parameter group and the following parameters within the group to configure your catalog types for notifications in SAP Asset Manager:

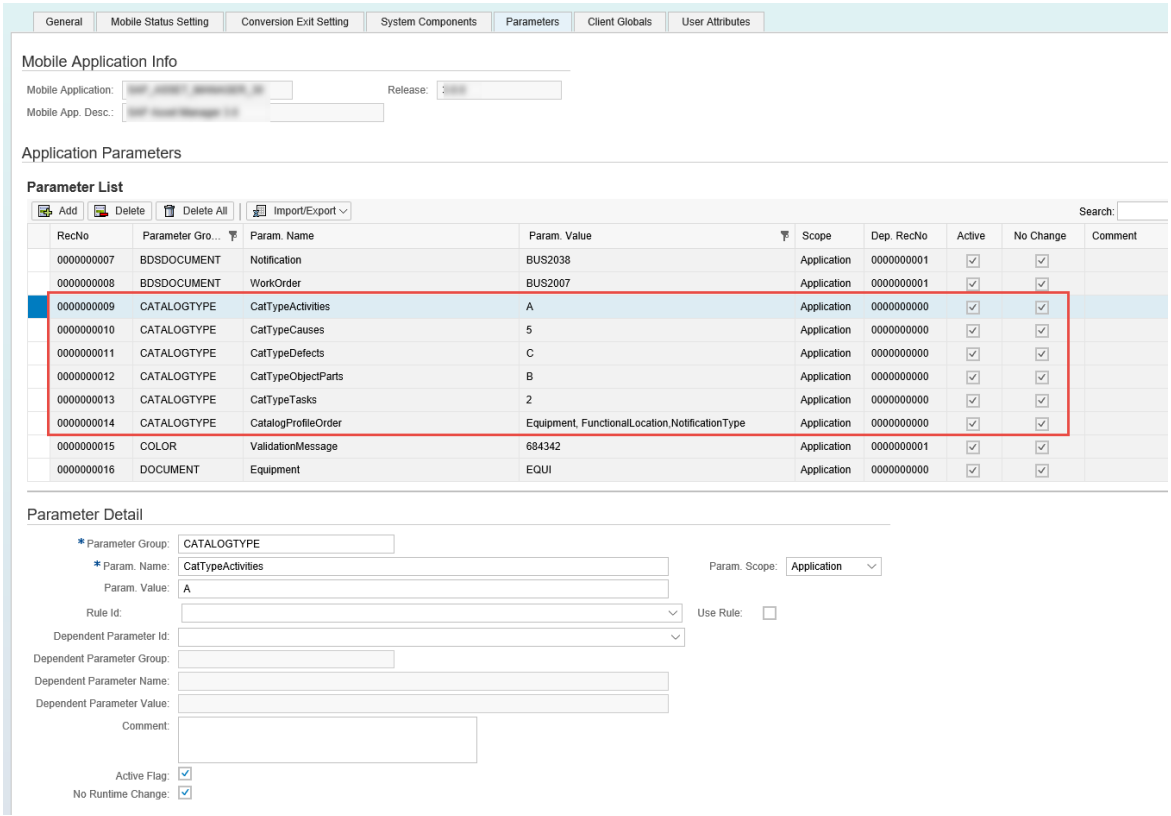
- **CatTypeActivities:** Default is *A*
- **CatTypeCauses:** Default is *5*
- **CatTypeDefects:** Default is *C*
- **CatTypeObjectParts:** Default is *B*
- **CatTypeTasks:** Default is *2*
- **CatalogProfileOrder:** Default is *Equipment, Functional Location, Notification Type*

The *CATALOGTYPE* parameters correspond to the rules found in the OData mobile data object *SAM1911_CATALOG_CODES*. You can add a new data filter rule to your customer namespace, or change the existing parameter-rule association to a new parameter-rule association.

Procedure

1. Using the ConfigPanel, navigate to *Mobile Application Configuration* > *Parameters tab*. In the left column, *Defined Mobile Applications*, select your application.

   The *Parameter List* populates with a list of all parameters available for the application.
2. The *CatType[xxx]* parameters are found in the *CATALOGTYPE* group. You can scroll down to find the parameter, or perform a search using the *Search* box. Highlight the parameter you want to configure and click the *Change* button.



3. Make your desired parameter association changes, or change the value of a parameter to *Z*, a custom activity catalog type.
4. Check the `<Active>` flag to ensure that the parameter is used by the mobile application. If desired, and if not already checked, check the `<No Runtime Change>` box to ensure that the value of the parameter is not overridden at runtime through synchronization processing.
5. *Save* your changes.
6. If you are creating a custom activity value type, navigate to *OData Mobile Data Object Configuration* > *Data Filter Tab* > *SAM1911_CATALOG_CODES* > *Operation - READ* > *Standard Filter* > *CATALOG_TYPE*.
7. Click the *Change* button. Add the new value. For information on working with rules, see Working with oData MDO Filter Rules [page 72].
8. *Save* your changes.

3.1.5 Enabling and Disabling Features Per User Through SAP Authorization

Using parameter framework configuration, configure parameters to enable or disable various features per the authorization of the user in the back end.

Each mobile user is connected to a back end SAP user. The back-end SAP user can be assigned one or more roles. These roles grant their holders authorizations within the back end system. Through parameter configuration, SAP provides a standard rule handler that performs a TCode authorization. SAP also provides new globals that can turn on and off new features.

If a parameter is enabled to use a rule instead of a global, and the user role has an authorization to run a specific transaction code, then that specific feature is enabled for that SAP user. If the user has the authorization' for a specific transaction code, then the specific feature is disabled for that mobile user. Therefore, depending on the authorization of the SAP user, the feature now either works or doesn't work, displays, or doesn't display (depending on the feature function), rather than is turned on or off for all users.

Features Available Through SAP Authorization

The following features are available for you to enable or disable starting with the SAP Asset Manager 1911 release. Use the following subsection to learn how to use the ConfigPanel to enable or disable a feature based on the authorization of the user.

Component	Functionality	Category	TCODE	Back-End Parameter	Comments
SAP ASSET MANAGER	Create work order	Work Orders	IW31	Enable.WO.Create	Includes operations and suboperations

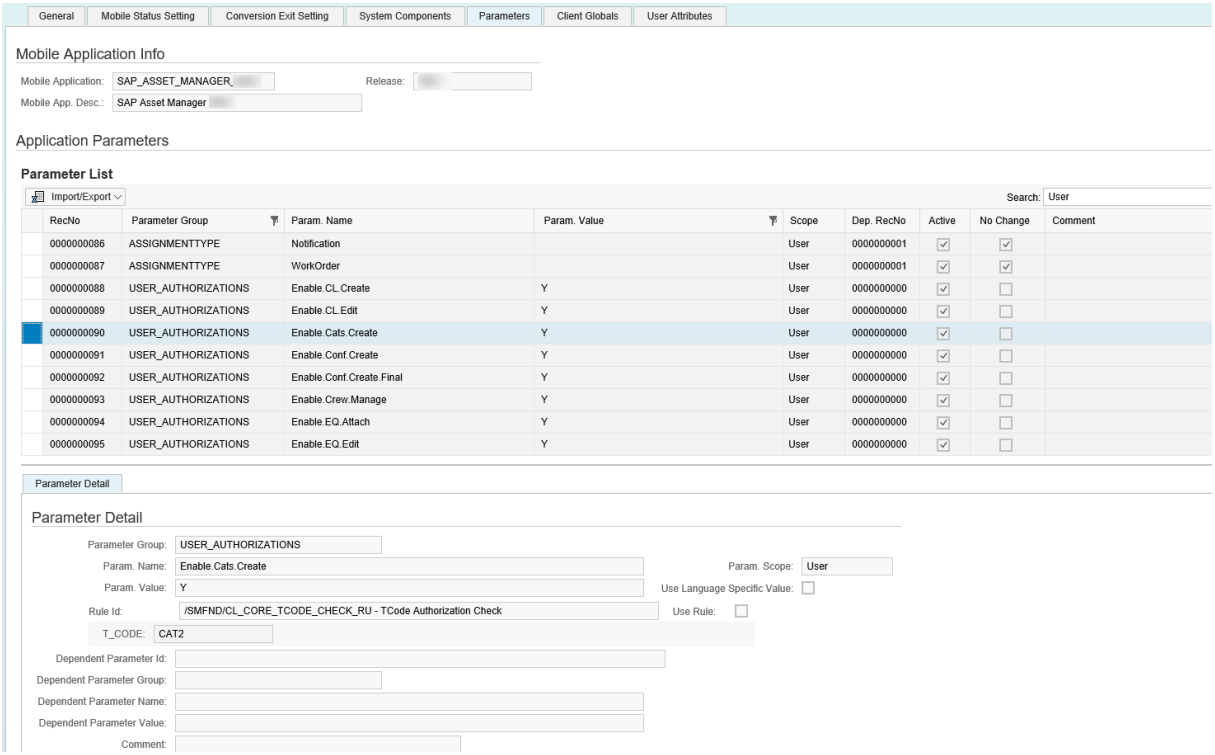
Component	Functionality	Category	TCODE	Back-End Parameter	Comments
SAP ASSET MANAGER	Edit work order	Work Orders	IW32	Enable.WO.Edit	Includes operations and suboperations (except local)
SAP ASSET MANAGER	Create Notification	Notifications	IW26	Enable.NO.Create	Includes items, tasks, and activities
SAP ASSET MANAGER	Edit Notification	Notifications	IW22	Enable.NO.Edit	Includes items, tasks, and activities (except local)
SAP ASSET MANAGER	Edit FLOC	Functional Location	IL02	Enable.FL.Edit	Includes adding characteristics (except local)
SAP ASSET MANAGER	Edit equip	Equipment	IE02	Enable.EQ.Edit	Includes adding characteristics, install, and dismantle (except local)
SAP ASSET MANAGER	Measurement readings	Measurement	IK11	Enable.MD.Create	
SAP ASSET MANAGER	Equip attachment upload	Attachments	N/A	Enable.EQ.Attach	See the *Generic Authorization Check* section in this topic
SAP ASSET MANAGER	FLOC attachment upload	Attachments	N/A	Enable.FL.Attach	See the *Generic Authorization Check* section in this topic
SAP ASSET MANAGER	Allow time recording	CATS	CAT2	Enable.Cats.Create	
SAP ASSET MANAGER	Allow confirmations	Confirmation	IW41	Enable.Conf.Create	
SAP ASSET MANAGER	Allow final confirmation	Confirmation	N/A	Enable.Conf.Create.Final	See the *Generic Authorization Check* section in this topic
SAP ASSET MANAGER	Issue and return parts	MIGO	N/A	Enable.Parts.Issue	See the *Generic Authorization Check* section in this topic
CUSTOMER SERVICE	Service notification create	Notifications	IW51	Enable.SNO.Create	
CUSTOMER SERVICE	Service notification edit	Notifications	IW52	Enable.SNO.Edit	Except local

Component	Functionality	Category	TCODE	Back-End Parameter	Comments
ASSET CENTRAL	Add checklist	Checklist	N/A	Enable.CL.Create	See the *Generic Authorization Check* section in this topic
ASSET CENTRAL	Fill checklist	Checklist	N/A	Enable.CL.Edit	See the *Generic Authorization Check* section in this topic
CREW	Manage crew	Crew	N/A	Enable.Crew.Manage	See the *Generic Authorization Check* section in this topic

How to Enable or Disable Features Per User Through Parameters

1. Using the ConfigPanel, navigate from the main screen to *Mobile Application Configuration* > *Parameters tab*. In the left column, *Defined Mobile Applications*, select your application.
   The *Parameter List* table populates with a list of all globals available for the application.
2. Perform a search for the parameter you want to enable or disable as a feature by user role by using the table in this topic to ensure that the parameter is available in the parameter framework for configuration. Search for your parameter in the *Parameter List* using the *Search* box. All user authorization parameters are found under the `<Parameter Group>` name of *USER_AUTHORIZATIONS*. Select your parameter and click the *Change* button.
3. The rule */SMFND/CL_CORE_TCODE_CHECK_RU - TCode Authorization Check* is already selected for you in the `<Rule ID>` field. When you check the `<Use Rule>` checkbox, the rule is active.
4. Change the `<Param. Scope>` dropdown selection from *Application* to *User*.
5. If needed, select the appropriate `<Dependent Parameter ID>` from the dropdown list.
6. Check the `<Active Flag>` checkbox to ensure that your new parameter is active for the user role. *Save* your changes.

See the following screenshot for an example of a configured user role parameter:



Generic Authorization Check

Use the rule */SMFND/CL_CORE_AUTH_CHECK_RU - TCode Authorization Check* to enable a more generic authorization check. This rule is used for the following parameters:

- Enable.EQ.Attach
- Enable.FL.Attach
- Enable.Conf.Create.Final
- Enable.Parts.Issue
- Enable.CL.Create
- Enable.CL.Edit
- Enable.Crew.Manage

To use the */SMFND/CL_CORE_AUTH_CHECK_RU - TCode Authorization Check*, do the following:

1. Create an authorization object. See Authorization Objects for more information.
2. Assign the authorization object to a role.
3. Assign the role to users who will be using the feature.
4. Configure the rule in the ConfigPanel to reference the new authorization object. For more information, see the AUTHORITY-CHECK topic.

3.1.6 Enabling and Disabling Local Object Mobile Status

The EnableOnLocalBusinessObjects parameter found in the MOBILESTATUS group enables or disables a mobile client user to take a local object through its entire lifecycle even if the client is offline.

Context

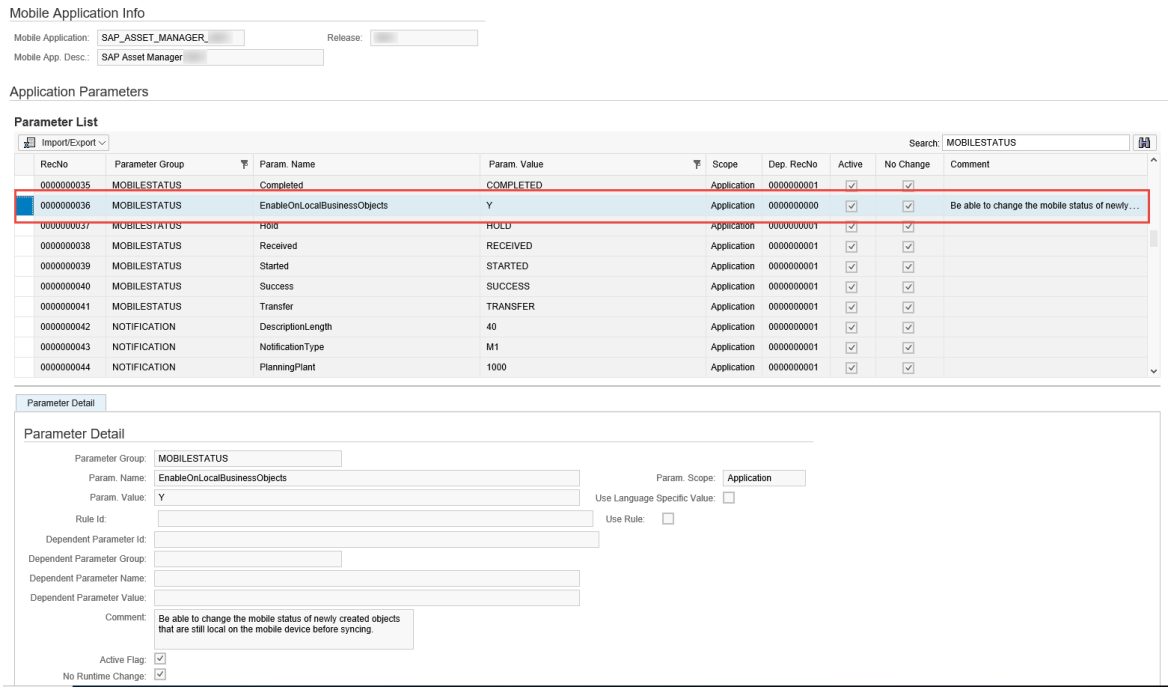
If you enable the *EnableOnLocalBusinessObjects* parameter, SAP Asset Manager allows mobile status update changes for the following:

- Work Order - Assignment level 1
- Operation - Assignment level 2
- Suboperation - Assignment level 3
- Notification
- Notification task
- Notification item task

Procedure

1. Navigate to *Mobile Application Configuration* > *Parameters tab* using the ConfigPanel. Select your application in the left column, *Defined Mobile Applications*.

   The *Parameter List* populates with a list of all parameters available for the application.
2. Find the *EnableOnLocalBusinessObjects* parameter in the *MOBILESTATUS* group. Scroll down to find the parameter, or perform a search using the *Search* box. Highlight the parameter you want to configure and click the *Change* button.



3. Enable or disable the parameter using the following strings: *Y*, *Yes*, *T*, or *True* are used to enable the parameter. *N*, *No*, *F*, or *False* are used to disable the parameter.
4. Check the `<Active>` flag to ensure that the parameter is used by the mobile application. Check the `<No Runtime Change>` box to ensure that the value of the parameter isn't overridden at runtime through synchronization processing if desired and if not already checked.
5. *Save* your changes.

3.2 Geospatial Service Definitions Procedures

3.2.1 Configuring Your Map Settings

Some GIS settings are standard with the initial SAP Asset Manager application.

You can change any of the settings described in this topic to configure the application for your site.

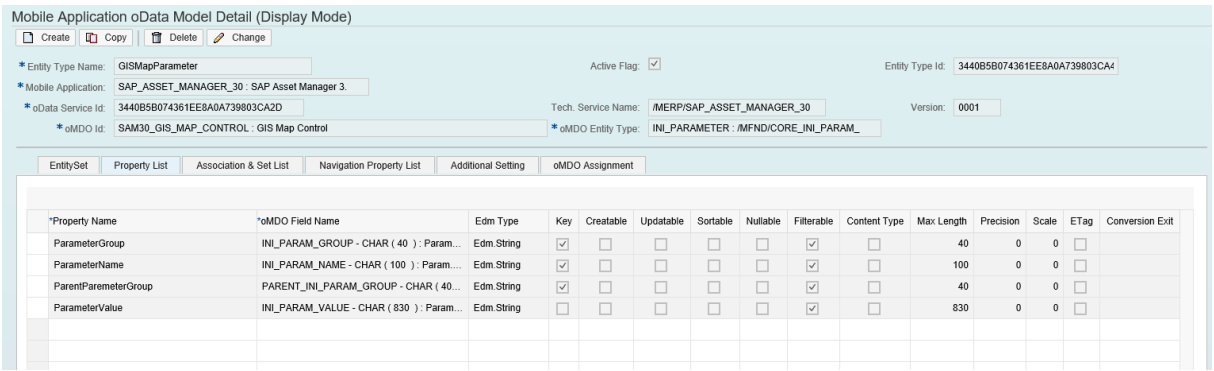
> **i Note**
>
> You can also change the map setting metadata through the Mobile Development Kit. Note that if there are metadata differences, Mobile Development Kit changes override ConfigPanel changes.

OData Model Configuration - Property List Tab

In the ConfigPanel, the *GISMapParameter* entity type contains the following properties:

- ParameterGroup
- ParameterName
- ParentParameterGroup
- ParameterValue

Use the fields in the following section to properly categorize these parameters.



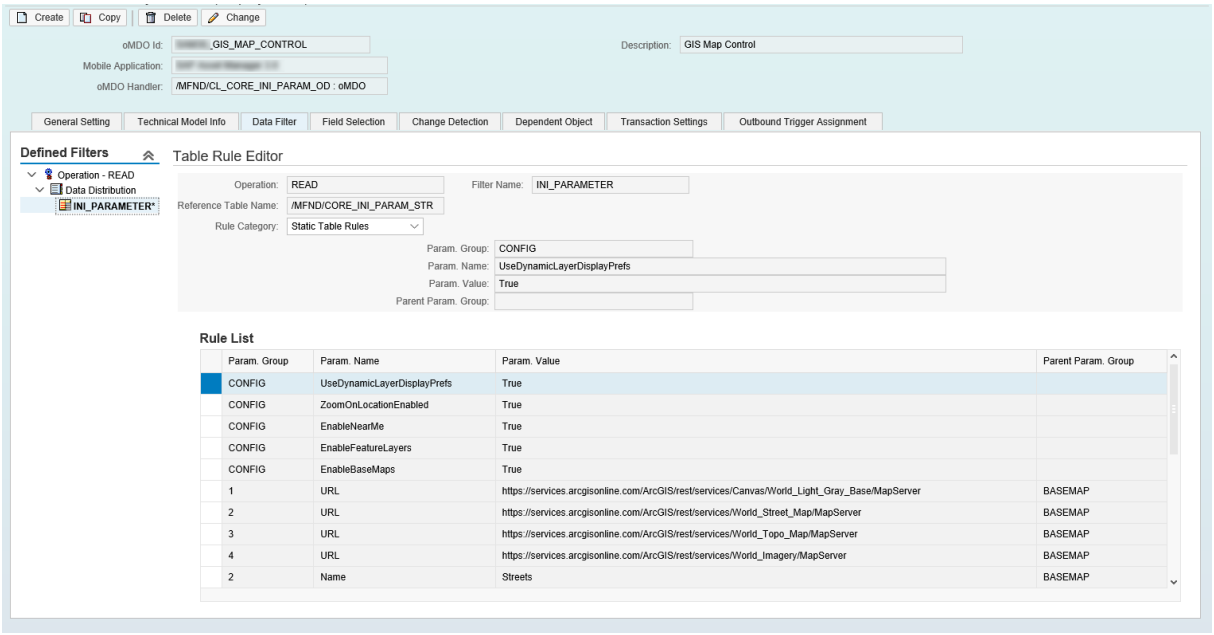
OData Mobile Data Object Configuration - Data Filter Tab

From the ConfigPanel Home page, navigate to *OData Mobile Data Object Configuration* > *Data Filter Tab* > *<SAMXX>_GIS_MAP_CONTROL* > *Operation - READ* > *Data Distribution* > . Click the *Change* button.

The following parameters are standard:

- **CONFIG:** Parameter Group
- **BASEMAP:** Parent Parameter Group
- **FEATURELAYER:** Parent Parameter Group

See the following for an example screenshot of the parameters in the ConfigPanel in the *Data Filter* tab, and a table representing how to configure the parameters in the tab.



Parameter Group	Parameter Name	Parameter Value
CONFIG	EsriClientID	

3.2.2 Supporting Authenticated GIS Services

Use

You can view token-based authenticated basemaps and feature layers on the mobile client. Use the ConfigPanel to configure the client ID and client secret strings.

Configuring Authenticated GIS Services

The mobile client retrieves the tokens. The client ID and client secret are supplied to the client so each client can generate their tokens for accessing authenticated services.

If your organization wishes to access Esri application-level authenticated GIS services, configure the SAP Asset Manager application as shown in the following procedure.

You can also configure a proxy through Esri. Authenticated basemaps and feature layers are requested through a local proxy. The proxy manages the generation and use of tokens based on the client ID and client secret. For more information on configuring a proxy, see the Esri documentation, Working with Proxy Services .

To turn on GIS authenticated services in the ConfigPanel, add the following rule:

1. From the ConfigPanel *Home* page, navigate to *OData Mobile Data Configuration* > *OData Mobile Data Object List* > *Data Filter Tab* > *SAP_ASSET_MANAGER_<XX>* > *SAM1911_GIS_MAP_CONTROL* > *Operation - READ* > *Data Distribution* > *INI_PARAMETER*.
2. Click the *Change* button. In the *Rule List* section, click the *Add* button to add a new rule. The rule gives you the freedom to retrieve your client credentials in a manner appropriate for your organization:
   - **Parameter Group:** `AUTHENTICATION`
   - **Parameter Name:** `ConfigRule`
   - **Parameter Value:** `/SAPAssetManager/Rules/Path/To/A/Rule.js`
     The specified rule must return a json object in the form of the following example. Be sure to replace the text in the sample code with your actual client ID and client secret (`"YourClientId"` and `"YourClientSecret"`):

     Sample Code

     ```
     {
         "ClientId": "YourClientId",
         "ClientSecret": "YourClientSecret"
     }
     ```

3.3 OData Channel Integration Settings Procedures

3.3.1 Copying an Object to the Customer Namespace

When you modify either an oData mobile data object or an exchange object, first make a copy of the object and place it in the customer namespace.

Context

The following procedure provides information on making a copy of an oData mobile data object (OMDO) or exchange object within SAP Mobile Add-On. In any of the procedures provided in this guide where an OMDO or

an exchange object is copied, refer to this procedure for instructions. When you copy either an OMDO or an exchange object, you can roll back any changes you make to the application if necessary without changing the original objects.

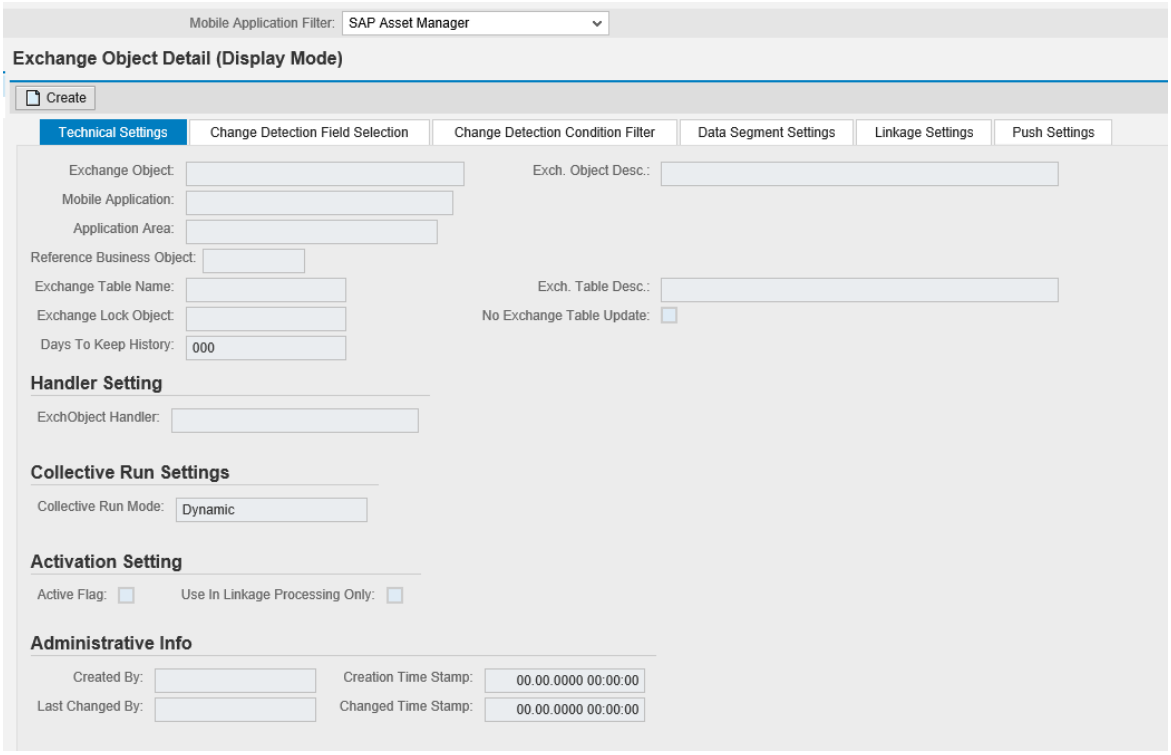
Once you copy an OMDO and modify the object, you may adjust the oData model definition to reference the new OMDO. Similarly, when you copy and modify an exchange object, you may need to change the EFI trigger assignment to the new exchange object. These procedures are covered separately.

Procedure

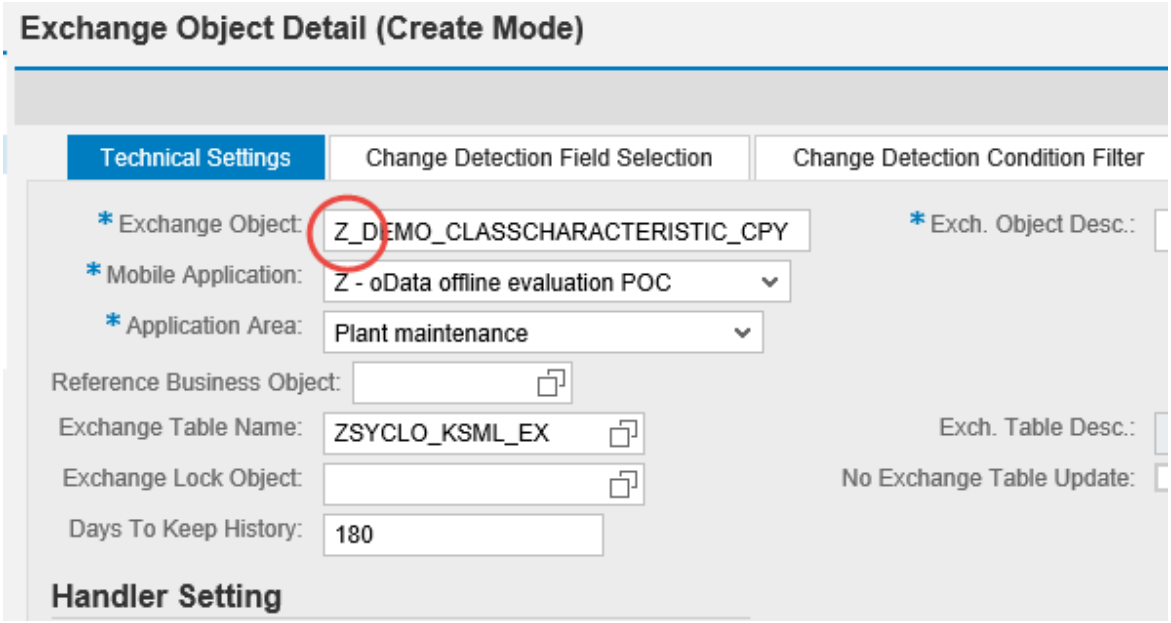
1. Log into the ConfigPanel of the SAP Mobile Add-On.
2. Click either *Exchange Object Configuration* or *oData Mobile Data Object Configuration* from the home page.

   The Object Detail panel opens.

   > **i Note**
   >
   > Figures shown in this procedure are taken from the Exchange Object configuration page. Screens may look different when configuring an oData mobile data object. For either, the ability to copy is provided.



3. Select the object to copy from the list of OMDOs or exchange objects and click *Copy*.
4. In the main object `<ID>` field, add a `<Z>` to the beginning of the object name.



5. Click *Save* to save the object copy.

   A copy of the original object is created in the customer namespace. Now you can modify the object, with the original object as a back-up for rollback purposes, if necessary.

3.3.2 Working with oData MDO Filter Rules

Filter rules specify a single field within the database tables from which data is retrieved. Filter rules also specify under which conditions records are included in the operation based on the value of the field.

Data filters are part of the configuration of an oMDO. If you make configuration changes to SAP Asset Manager, you may need to adjust the rules for one or more of the oMDO filters.

Many of the filters in SAP Asset Manager either do not contain active rules or contain rules that you can adjust. A filter only effects the synchronization behavior when it has one or more active rules.

The following procedure instructs you on how to adjust a filter using the ConfigPanel.

3.3.2.1 Changing oData MDO Filter Rules

Many of the common configuration changes made for an SAP Asset Manager implementation involve modifying or adding one or more filter rules in an oData MDO.

Context

In SAP S/4HANA, each user is assigned a role based profile with authorization permissions on viewable data and available activities. For example, a user working in one plant should not be able to view data for a different

plant. When business activities performed by a user are mobilized through the mobile application, the ability to extend the same restrictions to the mobile application is necessary. Data filter rules provide the function to restrict data access for mobile applications.

Use the following procedure to modify a data filter rule for an oMDO. The changes you make to the settings of a given rule vary depending on your mobile application implementation requirements. Subsequent procedures in the *Configuration Guide* refer to this procedure and provide detailed values and settings for filter rules involved in the specific change.

Procedure

1. Access the ConfigPanel through SAP Mobile Add-On.
2. From the ConfigPanel *Home* page, click the *oData Mobile Data Object Configuration* link.
3. At the top of the *oData Mobile Data Object Configuration* page display, in the *Mobile Application Filter* field, choose your mobile application from the dropdown menu. Choosing your mobile application is not a necessary step, but it eliminates objects that are not part of your mobile application from the object list.
4. Click the *Data Filter* tab.
5. Expand the *oData Mobile Data Object List* tree so you can see all of the oData mobile objects.
6. Select the oData mobile data object that requires filter modification from the list.

   The current rule filter settings are displayed in the *Rule Editor* section. All existing rules for the filter are displayed in the *Rule List* table.
7. To add a new rule, edit an existing rule, or delete a rule from the filter, click *Change*.

   Many of the fields in the rule editor become editable, and the buttons *Add Row* and *Delete Row* appear.
8. Set or modify any editable fields desired according to your mobile application needs. For a detailed description of all oData mobile data object fields, see and the related subtopics.
9. Set the *Active Flag* to `<True>` for each added or edited field before saving changes. Inactive filter rules have no effect on synchronization processing.
10. Click *Save* to apply your changes.

3.3.3 Work Order Distribution by Order Type Overview

In the default configuration of SAP Asset Manager, work orders are distributed to technicians based on basic parameters. Your site may wish to distribute work orders to users based on the order type.

By default, all Plant Maintenance specific order types are included in the synchronization logic for the SAP Asset Manager application.

In many environments, one or more order types are added to SAP Mobile Add-On specifically for work orders that are distributed to technicians. The added order types indicate that SAP Asset Manager will only download certain specified work orders. To support this distribution method, change the data filter rules of the OMDOs involved in work order synchronization. The OMDOs include:

- **SAM1911_ORDER_TYPE**
- **SAM1911_WORK_ORDER_GENERIC**

3.3.3.1 Configuring Work Order Distribution by Work Order Type

Creating rules based on work order types affects synchronization processing and work order downloads to the mobile devices of your users.

Prerequisites

Address the following before performing the procedure:

- The order types for work orders that are downloaded to technicians using the SAP Asset Manager application are already determined.
- The person performing the procedure has access to the Config Panel and permissions to change settings.

Context

The following procedure modifies the synchronizing behavior of the SAP Asset Manager application so only work orders with a given order type or types are downloaded to the client. In the procedure, you'll change the ORDER_TYPE filter in the OMDOs involved in work order synchronization. Specifically, you add rules to the filter in each OMDO to include only the desired work order types. You add a rule for each order type to include.

If you don't create a rule for a work order type, then those work order types are excluded from the work order download synchronization processing. If the work orders are excluded from the synchronization processing, then the work orders aren't present on the mobile clients of your users.

Procedure

1. From the Config Panel home page, click the *OData Mobile Data Object Configuration* link, then click the *Data Filter* tab. Be sure to have your desired mobile application chosen in the *Mobile Application Filter* field at the top of the page.
2. Expand the *OData Mobile Data Object by Mobile App* list on the left and click **SAM1911_ORDER_TYPE**.
3. Expand the *Standard Filter* in the *Defined Filters* pane, and click the *ORDER_TYPE* filter.
4. View the rule list for the filter, which is empty in the default configuration of SAP Asset Manager. Click the *Change* button.
5. Create a rule for each order type included in the work order distribution to the SAP Asset Manager technicians. The settings for the rule are as follows:
   - **DOF Rule Type:** Static Value in Range Format
   - **Sign:** Inclusive
   - **Option:** =
   - **Low Value:** The desired order type

   For more details on adding or editing filter rules, see Changing oData MDO Filter Rules [page 72].

6. *Save* your changes once you're finished.
7. Find and click the **SAM1911_WORK_ORDER_GENERIC** OData mobile data object on the list on the left.
8. Expand the *Operation - READ* > *Data Distribution* in the *Defined Filters* pane, and click the *ORDER_TYPE* filter.
9. View the rule list for the filter, which is empty in the default configuration of SAP Asset Manager. Click the *Change* button.
10. Create a rule for each order type included in the work order distribution to the SAP Asset Manager technicians, as you did with the previous OMDO filter. The settings for the rule are as follows:
    - **DOF Rule Type:** Static Value in Range Format
    - **Sign:** Inclusive
    - **Option:** =
    - **Low Value:** The desired order type
11. *Save* the changes.

Results

After you finish the procedure, work orders are downloaded by the SAP Asset Manager application only if their work order type is set to a type for which a filter rule was created. Other work order types aren't retrieved by the application.

3.3.4 Business Object Distribution by Assignment Model

Business object distribution defines the data that needs to be downloaded to the mobile device based on the resource planning of technicians for different business objects, such as work order and notification. You can use this configuration to define which technicians has to complete which activities on the mobile device.

Implementation environments in different business industries or business types may use a different business object model from the default to determine the proper technician assignment for a business object such work order and notification.

3.3.4.1 Work Order Assignment Type Options

By default, the SAP Asset Manager application determines the assignment of a work order using the personnel number of the work order header. However, you can make minor configuration changes to support several work assignment models.

For some customers using Assignment Type 3 for work orders, the ability to see a list of suboperations is more important than to see a list of operations. Work order headers are still visible. You can configure your preference using the ConfigPanel for SAP Asset Manager.

For assignment types 2 and 6, some customers may prefer the ability to view all operations rather than all work orders. Work order headers are still visible. You can configure your preference using the ConfigPanel for SAP Asset Manager.

Implementation environments in different business industries or business types may use a different business model from the default to determine the proper technician assignment for a work order.

The following assignment types are supported with minor configuration changes:

- **Assignment Type 1:** Header-level person responsible for the work order (default, no change required)
- **Assignment Type 2:** Operation-level personnel number of the work order
- **Assignment Type 3:** Sub-operation-level personnel number of the work order
- **Assignment Type 4:** Capacity requirement personnel assignment
- **Assignment Type 5:** Header-level planner group*
  Prerequisite: Mobile user has to have the user parameter *IHG* set up in the user profile parameter.
- **Assignment Type 6:** Operation- or task-level work center*
  Prerequisite: Mobile user has to have the user parameter *AGR* set up in the user profile parameter.
- **Assignment Type 7:** Header-level business partner*
- **Assignment Type 8:** Header-level of the work center*
  Prerequisite: Mobile user has to have the user parameter *VAP* set up in the user profile parameter.
- **Assignment Type A:** Multi resource scheduling (MRS)
- **Assignment Type Z:** Other (custom BAdI option)*

** These assignment types do not require a technician to have a personnel number.*

Follow the steps below to change the assignment type used in a deployment:

1. On the ConfigPanel home page, select *OData Mobile Data Object Configuration*. Make sure to select your desired mobile application in the *Mobile Application Filter* field at the top of the page.
2. In the *OData Mobile Data Object List* select **SAM1911_WORK_ORDER_GENERIC**, and then the *Data Filter* tab.
3. Expand the *Defined Filters* list as follows; *Operation - READ* > *Data Distribution* and click *WO_ASSIGNMENT_TYPE*. Click the *Change* button.
4. Set *Low Value* with the desired assignment type as defined by the assignment type model.
5. *Save* your changes.

> **i Note**
>
> If you're configuring an operation level assignment type, you must update the *OPER_EXCL_SYST_STAT* filter with the *I0009 - CNF:Confirmed* value. However, remove the *I0009 - CNF:Confirmed* value if you're configuring a header level assignment type.

3.3.4.2 Notification Assignment Options

By default, the SAP Asset Manager application determines the assignment of a notification associated with the notification header. However, you can make minor configuration changes to support several other assignment models for the notification object.

The following assignment types are supported for the notification object:

- **Assignment Type 1:** Header-level person responsible for the work order (default, no change required)
- **Assignment Type 2:** Task-level personnel number of the work order
- **Assignment Type 3:** Header-level planner group*
  Prerequisite: Mobile user has to have the user parameter *IHG* set up in the user profile parameter.
- **Assignment Type 4:** Header-level business partner*
- **Assignment Type 5:** Header-level of the work center*
  Prerequisite: Mobile user has to have the user parameter *VAP* set up in the user profile parameter.
- **Assignment Type D:** Dependent Queue
  By default, this assignment is based on the technician's work order dependent collection*.
- **Assignment Type Z:** Other (custom BAdI option)*

** These assignment types do not require a technician to have a personnel number.*

Follow the steps below to change the assignment type used in a deployment:

1. On the ConfigPanel home page, select *OData Mobile Data Object Configuration*. Make sure to select your desired mobile application in the *Mobile Application Filter* field at the top of the page.
2. In the *OData Mobile Data Object List* select **SAM1911_NOTIFICATION_GENERIC**, and then the *Data Filter* tab.
3. Expand the *Defined Filters* list as follows; *Operation - READ* > *Data Distribution* > and click *NOTIF_ASSIGNMENT_TYPE*. Click the *Change* button.
4. Set *Low Value* with the desired assignment type as defined by the assignment type model.
5. *Save* your changes.

3.3.5 Filtering Equipment Records by Equipment Status

A large set of records could affect performance on the SAP Asset Manager client. Therefore, you can employ more filtering based on the status of equipment.

By default, SAP Asset Manager filters records through a user-dependent rule based on the planning plant of the user.

To filter records on the status of equipment retrieved for the table stored on the SAP Asset Manager client, modify the **SAM1911_EQUIPMENT** OMDO. Specifically, in the following procedure, you will configure the *EQUI_INCL_SYS_STAT* filter with a rule that specifies which status or statuses to include. After you configure the rule, only the equipment records with the specified statuses are retrieved by the application for download to the clients.

A common equipment status is *INST*. However, the INST status is only one example of many options. You can configure other filters, either with this example, or in place of it.

For your given SAP Asset Manager implementation, thoroughly review the equipment data stored in the database before deciding which filter rules to configure. After your equipment review, create the appropriate filters within the **SAM1911_EQUIPMENT** OMDO.

3.3.5.1 Configuring an Equipment Status Filter for an Equipment Table

Prerequisites

Address the following items before performing the procedure:

- Know the status or statuses that you are filtering on for equipment synchronization, as they are used in the procedure
- Have access to the ConfigPanel and permissions to change configuration settings

Context

Use the following procedure to create a filter rule for the OMDO, **SAM1911_EQUIPMENT**. Specifically, you are adding a rule to the filter *EQUI_INCL_SYST_STAT*. After you add the filter rule, only the equipment records that match the ones configured in the rule are downloaded to the SAP Asset Manager client.

Procedure

1. Navigate to *ConfigPanel Home* > *OData Mobile Data Object Configuration*.
2. Select your application in the *Mobile Application Filter* field.

   Selecting an application filters the *OData Mobile Data Object by Mobile App* choices in the left panel with only OMDOs available in your application.
3. View the new OMDO copy by selecting it in the *OData Mobile Data Object by Mobile App list*.
4. Select the *Data Filter* tab.
5. In the *Defined Filters* list, click the *Operation - READ* > *Standard Filter* > *EQUI_INCL_USER_STAT* node.
6. Add a rule to the filter with the following configuration settings:
   - **Filter Rule Type:** Static Value in Range Format
   - **Sign:** Inclusive
   - **Option:** =
   - **Low Value:** Equipment status to filter on
   - **Active Flag:** Checked
7. Repeat the previous step to include additional statuses in the filter.
8. *Save* your changes.

Results

When you finish the procedure, the equipment records downloaded by the SAP Asset Manager application are filtered to only include records with the status or statuses configured in the filter rules.

Next Steps

You may need to filter equipment according to additional criteria. Test that the status filters created during this procedure are performing as expected before creating additional filters for the same data set. Regardless of additional changes, test the synchronization of the equipment data thoroughly after you modify the application.

3.3.6 Retrieving Additional Data for OData Mobile Data Objects

The default implementation of SAP Asset Manager includes the typical data values required by most users and at most implementation. However, it is a common requirement that additional values are retrieved and stored.

3.3.6.1 Adding New Retrievable Values for OData Mobile Data Objects

Prerequisites

Address the following before performing this procedure:

- Determine and note the field values as well as any table values you want to add, as well as which tables the desired fields reside in SAP Mobile Add-On
- You must have access to the ConfigPanel and permissions to change configuration settings within it

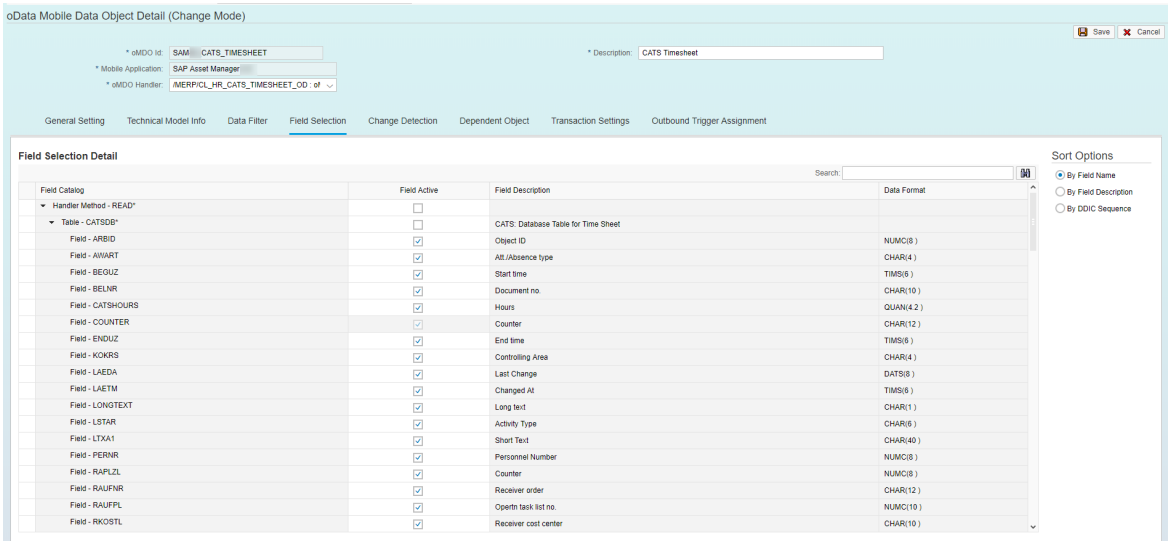
Context

Use the following procedure to add new fields to OData mobile data objects.

Procedure

1. Navigate to *ConfigPanel Home* > *OData Mobile Data Object Configuration*. Select the desired OMDO from the list on the left of the current configuration page.
2. Click the *Field Selection* tab, then click the *Change* button.

   The *Field Active* column is enabled.



3. In the *Field Selection Detail* pane, expand the *Handler Method* and the *Table* to search for the fields that you wish to activate. Fields that are already active are marked with a check in the checkbox.
4. Check the fields that you wish to activate and uncheck any active fields you wish to deactivate. *Save* your changes when finished.

Results

After completing the procedure, one or more new values are retrieved as part of the data for the object. The new values are displayed, edited, searched on, or used in other manners on the mobile client.

In the example screenshot in the procedure, the OData mobile data object used is **SAM1911_CATS_TIMESHEET**. To make other OMDO configuration changes to the object, navigate to the ConfigPanel home page, then click the *OData Model Configuration* link. On the left panel, find the corresponding *EntityType* to make any additional configuration changes. In this procedure example, the entity type is *CatsTimesheet*. See Setting up an OData Mobile Data Object [page 100] for more information.

3.3.7 Maintaining Customer Service Order Types

Prerequisites

Be sure that you have installed the Customer Service component. See the instructions in the Asset Manager Component Installation Guide for IOS for more information.

> **i** Note
>
> Configuring Customer Service order types is optional and is required only if the Customer Service component is enabled.

Procedure

1. Access the ConfigPanel through SAP Mobile Add-On.
2. From the ConfigPanel *Home* page, click the *OData Mobile Data Object Configuration* link.
3. At the top of the *OData Mobile Data Object Configuration* page display, in the *Mobile Application Filter* field, choose your mobile application from the dropdown menu. Choosing your mobile application is not a necessary step, but it eliminates objects that are not part of your mobile application from the object list.
4. Click the *Data Filter* tab.
5. Expand the *OData Mobile Data Object List* tree so you can see all of the OData mobile objects.
6. Select *SAM1911_WORK_ORDER_GENERIC* from the list. Then select *Data Filter tab* > *Operation - READ* > *Data Distribution* > *ORDER_TYPE*

   The current rule filter settings are displayed in the *Rule Editor* section. All existing rules for the filter are displayed in the *Rule List* table.
7. To activate the Customer Service order type, click *Change*.

   Many of the fields in the rule editor become editable.
8. Set the *Active Flag* to *True* for the `<Rule Value>` */MERP/CL_PM_ORDER_TYPE_ORU?CS*.
9. To apply your changes, click *Save*.

Results

The Customer Service order type is activated in the ConfigPanel.

3.3.8 Maintaining Customer Service Notification Types

Prerequisites

Be sure that you have installed the Customer Service component. See the instructions in the Asset Manager Component Installation Guide for IOS for more information.

> **i** Note
>
> Configuring Customer Service notification types is optional and is required only if the Customer Service component is enabled.

Procedure

1. Access the ConfigPanel through SAP Mobile Add-On.
2. From the ConfigPanel *Home* page, click the *OData Mobile Data Object Configuration* link.
3. At the top of the *OData Mobile Data Object Configuration* page display, in the *Mobile Application Filter* field, choose your mobile application from the dropdown menu. Choosing your mobile application is not a necessary step, but it eliminates objects that are not part of your mobile application from the object list.
4. Click the *Data Filter* tab.
5. Expand the *OData Mobile Data Object List* tree so you can see all of the OData mobile objects.
6. Select *SAM1911_NOTIFICATION_GENERIC* from the list. Then select *Data Filter tab* > *Operation - READ* > *Data Distribution* > *NOTIF_TYPE*

   The current rule filter settings are displayed in the *Rule Editor* section. All existing rules for the filter are displayed in the *Rule List* table.
7. To activate the Customer Service order type, click *Change*.

   Many of the fields in the rule editor become editable.
8. Set the *Active Flag* to *True* for the `<Rule Value>` */MERP/CL_PM_NOTIF_TYPE_ORU?CS*.
9. To apply your changes, click *Save*.

Results

The Customer Service notification type is activated in the ConfigPanel.

3.4 Push Framework Settings Procedures

3.4.1 Configuring Push for Work Order Assignment Type 1

Context

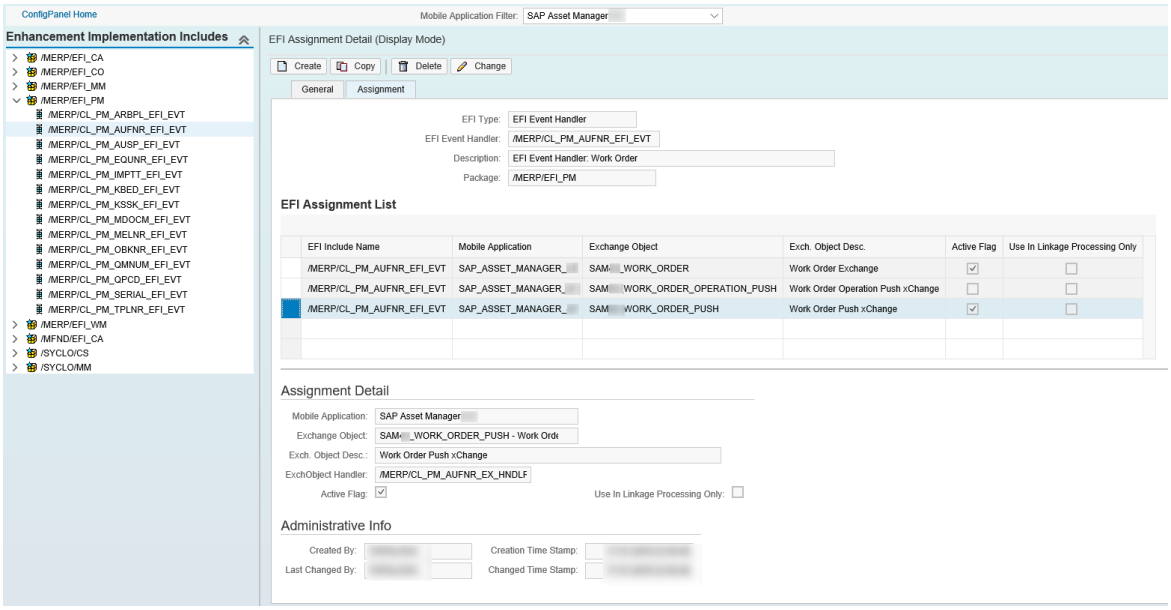
> **i** Note
>
> Push configuration is available for SAP Asset Manager for Android starting with the 4.0 release.

Event-based push is supported for assignment types 1 (header-level person responsible for the work order) and 2 (operation-level personnel number of the work order). You can only configure push for one work order assignment type at a time.

By default, work order push is enabled for whichever assignment type your work order OMDO is set to. You can manually assign the *WO_ASSIGNMENT_TYPE* filter for your data distribution model to *1* or *2* to set a push assignment type different to the assignment type of the work order data distribution.

Procedure

1. In the ConfigPanel, navigate to the *OData Mobile Data Object Configuration* section and find OMDO **SAM1911_WORK_ORDER_GENERIC**. Ensure the filter *WO_ASSIGNMENT_TYPE* is set to *1* on the *Data Filter* tab.
2. Return to the *Home* page of the ConfigPanel. Click the *Push Scenario Definition* link. Ensure that your mobileapplication is selected in the *Mobile Application Filter*.
3. Ensure the *Active* flag for the **SAM1911_EMERGENCY_WORKORDER_PUSH** scenario on the *General Data* tab is checked. Note that you can have both work order and notification pushes marked as Active because they are separate objects.
4. Make sure the configuration in the *Source Setting* and *Distribution Setting* sections are correct. By default, the `<Source Object>` for the work order operation push is the exchange object **SAM1911_WORK_ORDER_PUSH** with the `<Distribution Object>` **SAM1911_WORKORDER_PUSH**.
5. Return to the ConfigPanel Home page, then navigate to the *EFI Assignment* section. In the *Enhancement Implementation Includes* list, select *⁄MERP⁄EFI_PM* > *⁄MERP⁄CL_PM_AUFNR_EFI_EVT*.
6. Click the *Assignment* tab. Ensure the *Active* checkbox is checked for the exchange object **SAM1911_WORK_ORDER_PUSH**.



7. *Save* your changes.

3.4.2 Configuring Push for Work Order Assignment Type 2

Context

> **i Note**
>
> Push configuration is available for SAP Asset Manager for Android starting with the 4.0 release.

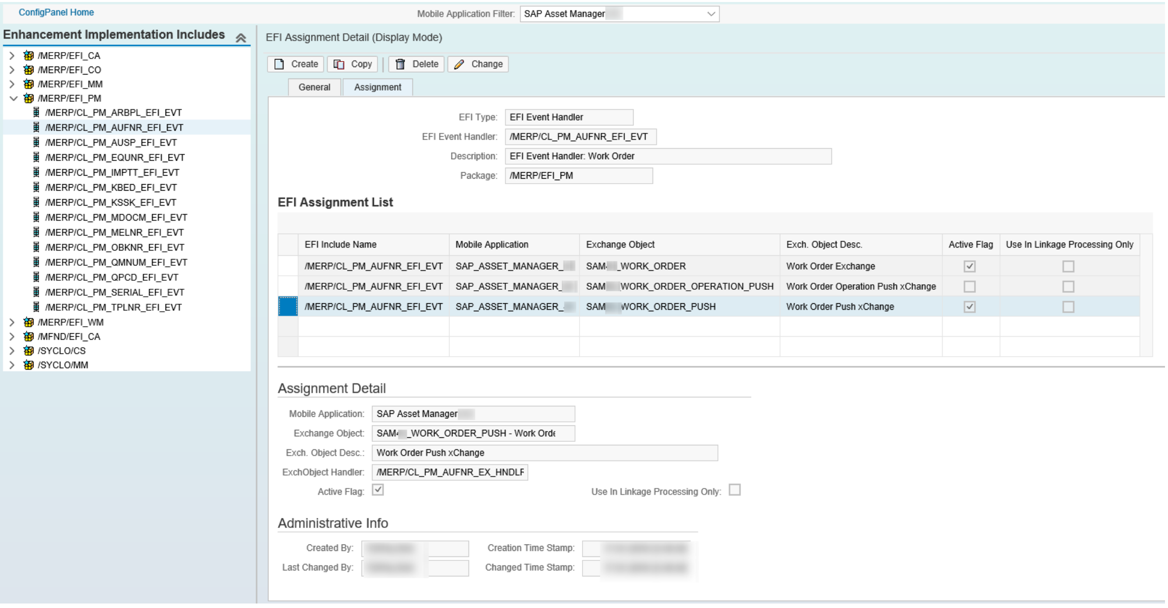
Event-based push is supported for assignment types 1 (header-level person responsible for the work order) and 2 (operation-level personnel number of the work order). You can only configure push for one work order assignment type at a time.

By default, work order push is enabled for whichever assignment type your work order OMDO is set to. You can manually assign the *WO_ASSIGNMENT_TYPE* filter for your data distribution model to *1* or *2* to set a push assignment type different to the assignment type of the work order data distribution.

Procedure

1. In the ConfigPanel, navigate to the *OData Mobile Data Object Configuration* section and find OMDO **SAM1911_WORK_ORDER_GENERIC**. Ensure the filter *WO_ASSIGNMENT_TYPE* is set to *2* on the *Data Filter* tab.
2. Return to the *Home* page of the ConfigPanel. Click the *Push Scenario Definition* link. Ensure that your mobileapplication is selected in the *Mobile Application Filter*.

3. Ensure the *Active* flag for the **SAM1911_EMERGENCY_WORKORDER_PUSH** scenario on the *General Data* tab is checked. Note that you can have both work order and notification pushes marked as Active because they are separate objects.
4. Make sure the configuration in the *Source Setting* and *Distribution Setting* sections are correct. By default, the `<Source Object>` for the work order operation push is the exchange object **SAM1911_WORK_ORDER_PUSH** with the `<Distribution Object>` **SAM1911_WORKORDER_PUSH**.
5. Return to the ConfigPanel Home page, then navigate to the *EFI Assignment* section. In the *Enhancement Implementation Includes* list, select *▶ /MERP/EFI_PM ▶ /MERP/CL_PM_AUFNR_EFI_EVT ▶*.
6. Click the *Assignment* tab. Ensure the *Active* checkbox is checked for the exchange object **SAM1911_WORK_ORDER_OPERATION_PUSH**.



7. *Save* your changes.

3.4.3 Configuring Push for Notification Assignment Types 1 - 5

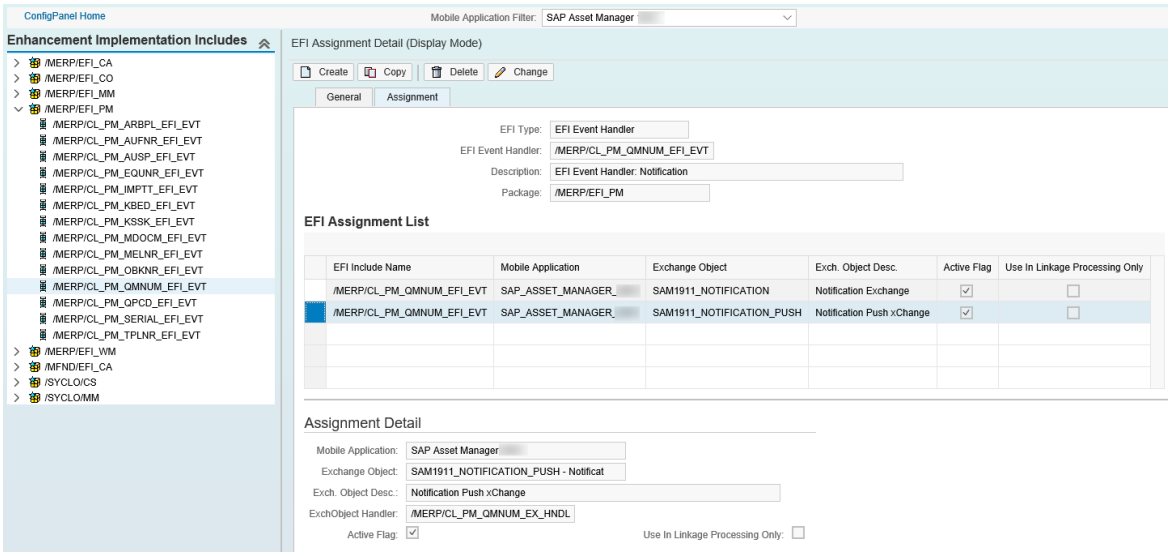
Context

Event-based push is supported for notification assignment types 1 through 5. You can only configure push for one notification assignment type at a time.

By default, notification push is enabled for whichever assignment type your notification OMDO is set to. You can manually assign the *NOTIF_ASSIGNMENT_TYPE* filter for your data distribution model to *1*, *2*, *3*, *4* or *5* to set a push assignment type different to the assignment type of the notification data distribution.

Procedure

1. In the ConfigPanel, navigate to the *OData Mobile Data Object Configuration* section and find OMDO **SAM1911_NOTIFICATION_GENERIC**. Ensure the filter *NOTIF_ASSIGNMENT_TYPE* is set to the assignment type of your choice (*1 - 5*) on the *Data Filter* tab.
2. Return to the *Home* page of the ConfigPanel. Click the *Push Scenario Definition* link. Ensure that your mobile application is selected in the *Mobile Application Filter*.
3. Ensure the *Active* flag for the **SAM1911_EMERGENCY_NOTIFICATION_PUSH** scenario on the *General Data* tab is checked. Note that you can have both work order and notification pushes marked as Active because they are separate objects.
4. Make sure the configuration in the *Source Setting* and *Distribution Setting* sections are correct. By default, the `<Source Object>` for the notification push is the exchange object **SAM1911_NOTIFICATION_PUSH** with the `<Distribution Object>` **SAM1911_NOTIFICATION_PUSH**.
5. Return to the ConfigPanel Home page, then navigate to the *EFI Assignment* section. In the *Enhancement Implementation Includes* list, select */MERP/EFI_PM* > */MERP/CL_PM_QMNUM_EFI_EVT*.
6. Click the *Assignment* tab. Ensure the *Active* checkbox is checked for the exchange object **SAM1911_NOTIFICATION_PUSH**.



7. *Save* your changes.

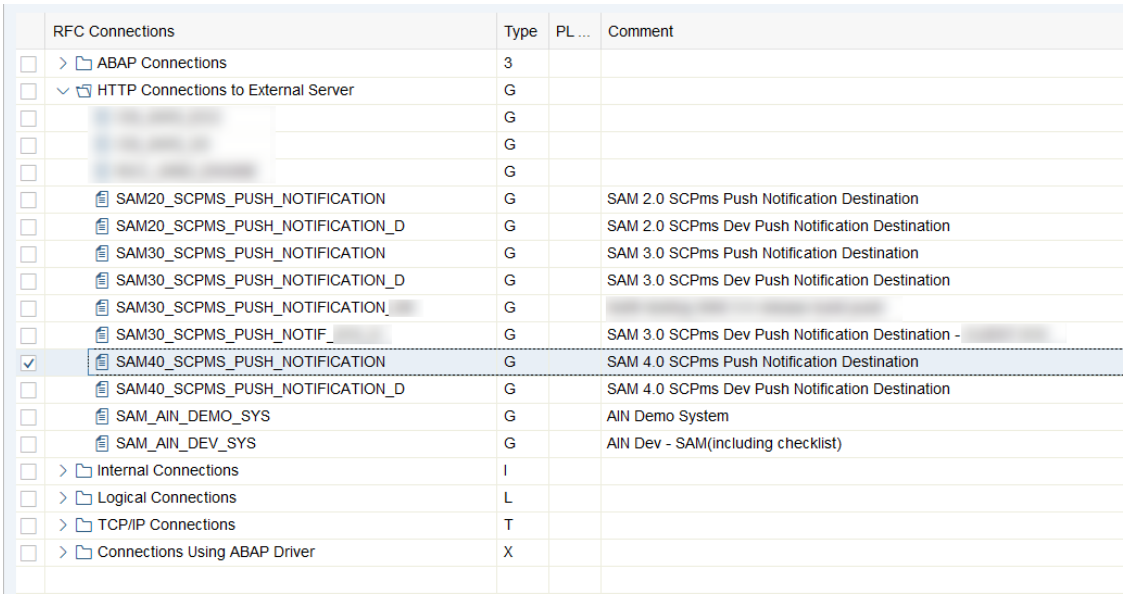
3.4.4 Setting up the Outbound Trigger for your Push Configuration

Context

> **i Note**
>
> Push configuration is available for SAP Asset Manager for Android starting with the 4.0 release.

Procedure

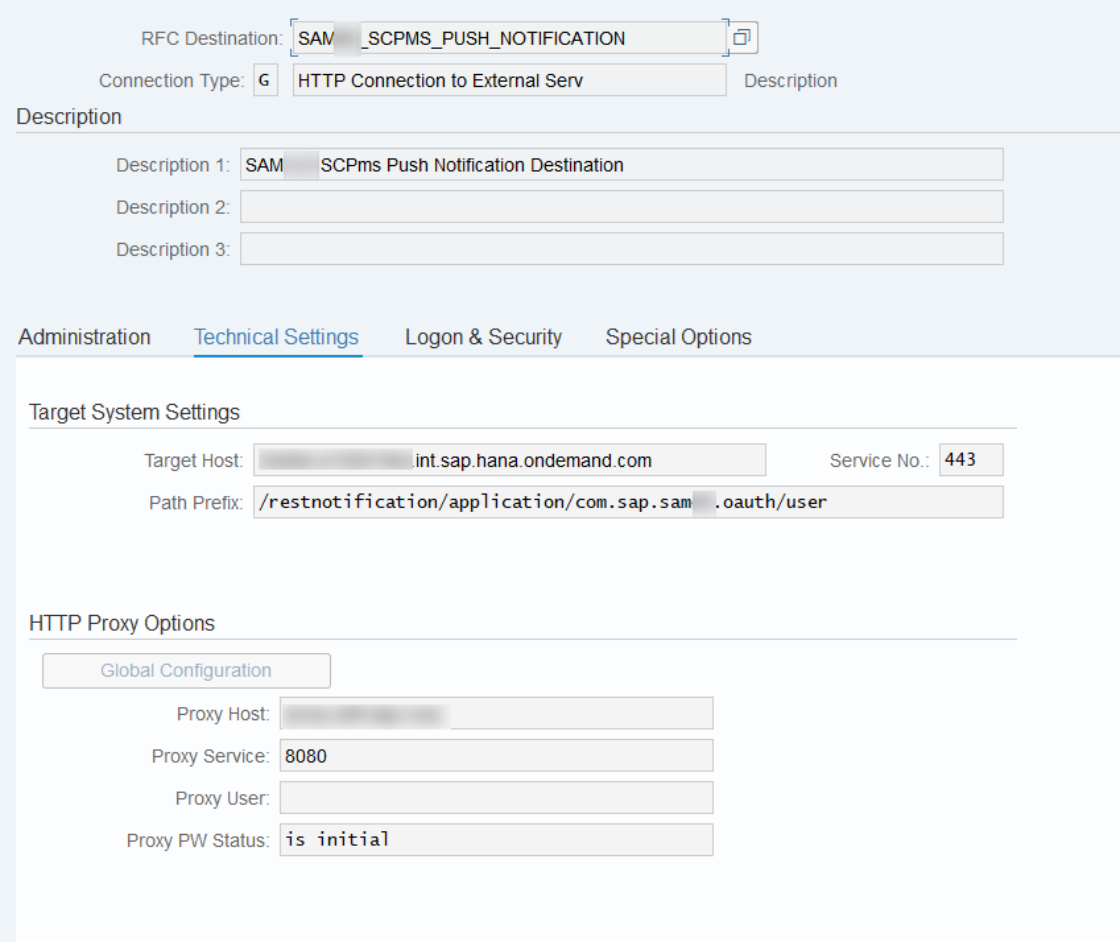
1. From the ConfigPanel *Home* page, navigate to the *Outbound Trigger Configuration* section and select your desired mobile application from the *Mobile Application Filter* dropdown menu at the top of the page.
2. From the *Outbound Triggers by Mobile App* list, select the outbound trigger **SAM1911_WORKORDER_TRIGGER_SCPMS**. Make sure that the `<Cloud Platform Mobile App ID>` matches your mobile services application ID from SAP Cloud Platform mobile services. By default, the application ID is set to *com.sap.<appXX>.oauth.prod*.
3. Set up the RFC destination **SAM1911_SCPMS_PUSH_NOTIFICATION** pointing to the mobile services host name using the SAP GUI:
   a. In the SAP GUI, using transaction *SM59*, add the following new RFC destination: **SAM1911_SCPMS_PUSH_NOTIFICATION** of type *G (HTTP Connection to External Serv)*



RFC Connections	Type	PL ...	Comment
ABAP Connections	3		
HTTP Connections to External Server	G		
[redacted]	G		
[redacted]	G		
[redacted]	G		
SAM20_SCPMS_PUSH_NOTIFICATION	G		SAM 2.0 SCPms Push Notification Destination
SAM20_SCPMS_PUSH_NOTIFICATION_D	G		SAM 2.0 SCPms Dev Push Notification Destination
SAM30_SCPMS_PUSH_NOTIFICATION	G		SAM 3.0 SCPms Push Notification Destination
SAM30_SCPMS_PUSH_NOTIFICATION_D	G		SAM 3.0 SCPms Dev Push Notification Destination
SAM30_SCPMS_PUSH_NOTIFICATION[redacted]	G		[redacted]
SAM30_SCPMS_PUSH_NOTIF_[redacted]	G		SAM 3.0 SCPms Dev Push Notification Destination - [redacted]
SAM40_SCPMS_PUSH_NOTIFICATION	G		SAM 4.0 SCPms Push Notification Destination
SAM40_SCPMS_PUSH_NOTIFICATION_D	G		SAM 4.0 SCPms Dev Push Notification Destination
SAM_AIN_DEMO_SYS	G		AIN Demo System
SAM_AIN_DEV_SYS	G		AIN Dev - SAM(including checklist)
Internal Connections	I		
Logical Connections	L		
TCP/IP Connections	T		
Connections Using ABAP Driver	X		

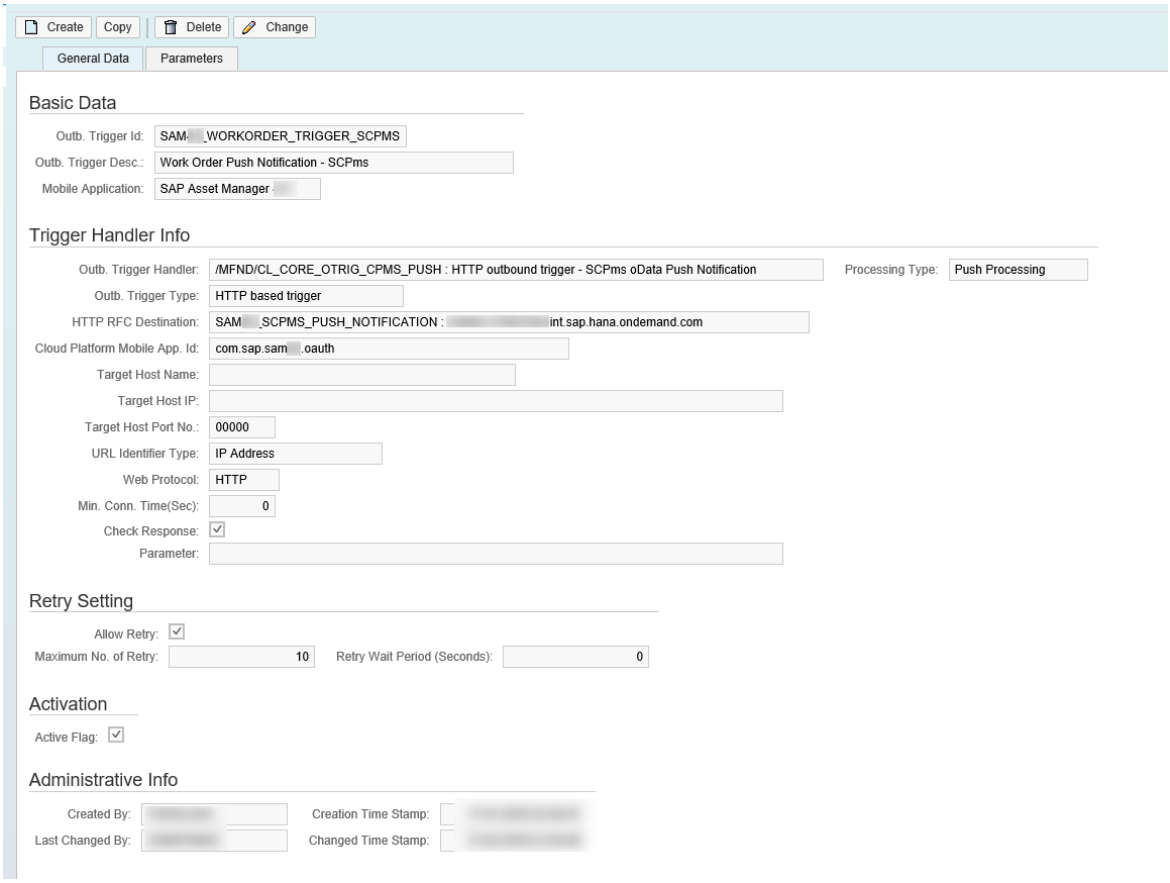
   b. On the *Technical Settings* tab of the new connection, set the *Target Host* to match the push API of the SAP Cloud Platform mobile services. Use service number *443*, which is the port number of the HTTPS connections.

> **i Note**
>
> If necessary, configure the proxy that you are using to allow your back-end systems to connect to the Internet.



c. On the *Logon & Security* tab, under the *Logon Procedure*, select *Basic Authentication*. Enter the user name and password of the service user. Assign the *Notification User* role to this service user in the SAP Cloud Platform mobile services configuration to ensure that the user is allowed to utilize the push notification service.
d. In the *Security Options* section of the *Logon & Security* tab, ensure that the *SSL* is set to *Active*.
e. In the *Security Options* section, ensure that the *SSL Certificate List* contains the SAP Cloud Platform certificate chain. You can check the certificate chain at the SSL client certificate list with the *STRUST* transaction. Display the SAP Cloud Platform mobile services certificate chain in any web browser by opening the SAP Cloud Platform mobile services cockpit and checking the security settings.
f. Save the connection and perform a connection test. If the configuration is completed properly, a 200 HTTP response is returned.
g. In the *Technical Settings* tab, add the rest of the path for the PUSH API in *Path Prefix* to allow the POST operation to work properly.

4. In the ConfigPanel, return to the *Outbound Trigger Configuration* panel, *General Data* tab, if you are not already there. Check the outbound trigger configuration to ensure that the previously established *HTTP RFC Destination* is set as the RFC destination to the proper mobile application. This configuration ensures that push notifications reach the right mobile application user.



5. On the *Parameters* tab, set parameter *SCPMS_WITH_SAP_USER_ID* to *True* if the SAP Cloud Platform user store is the same as the back end user store, that is, if you use the same mobile user for both SAP Cloud Platform logon and back end logon. If the user store is not identical, set the parameter to *False* and update the mobile user setting as explained in the next step.
6. If parameter *SCPMS_WITH_SAP_USER_ID* is set to *False*, maintain the user mapping under *User Management* using the Admin portal in the SAP GUI:

> **i Note**
>
> On the first delta sync, the SAP Asset Manager client automatically performs substeps *a-d* for you. If desired, you can still perform these substeps to verify that the push registration process has completed successfully.

   a. Using the SAP GUI, launch the Admin portal with transaction code */n/SYCLO/ADMIN*. On the Admin portal home page, select *Administration* > *User Management*. Make sure to select your desired mobile application in the *Mobile Application Filter* field at the top of the page. Choose *Search* to list all users for that application.
   b. Select *User ID* under *Search Result*, and click the *Client Registration Info* tab under the *Mobile User Detail* section. Choose *Change* from the menu bar.
   c. Enter the matching *CPms User Id* (using upper case) for the back end user name listed under this tab.
   d. *Save* your changes.
7. Return to the *Home* page of the ConfigPanel. Select the *Push Scenario Definition* page. Under *Push Scenarios by Mobile App* list, select the desired push scenario definition. Click the *Outbound Trigger* tab and ensure that the proper outbound trigger is assigned and active for the push scenario.

8. *Save* your changes.

3.4.5 Activating Default Push Services for SAP Asset Manager

Prerequisites

> **i** Note
>
> The *SAP Asset Manager Installation Guide* is a guide to setting up the basic framework necessary for push services using the default settings. For more details regarding configuration of push services, see the Push Scenario Definition [page 46] topic.

Before performing the procedure, ensure the following:

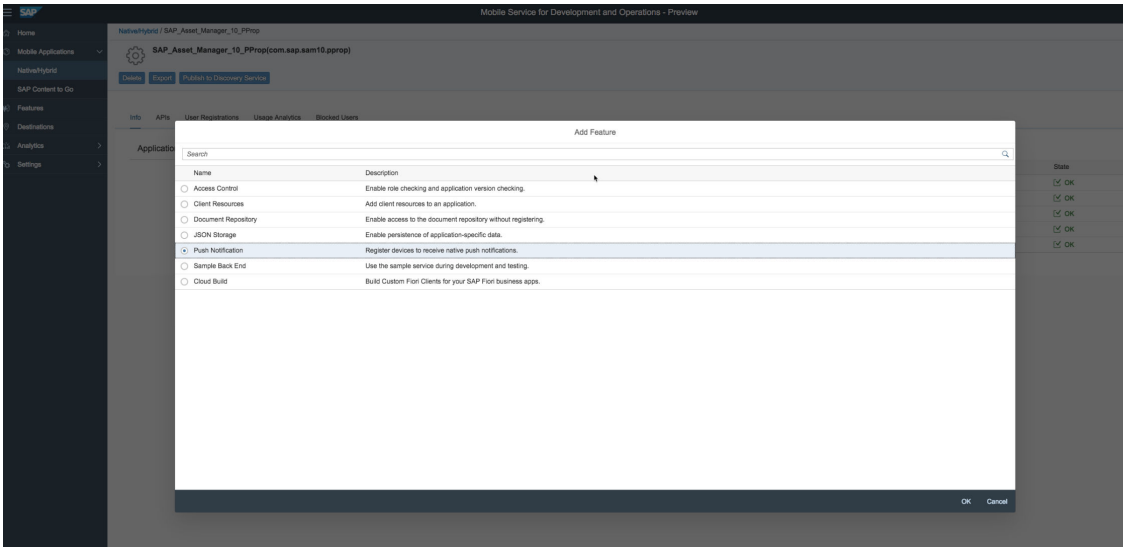
- SAP Asset Manager application on the device is running on Mobile Development Kit 2.2.001
- You have installed SAP Asset Manager 1911
- You have installed either SAP Mobile Add-On ECC or SAP Mobile Add-On for SAP S/4HANA. See the following installation guides on the following portal pages for version information:
  - SAP S/4HANA Mobile Add-On
  - SAP Mobile Add-On ECC

Context

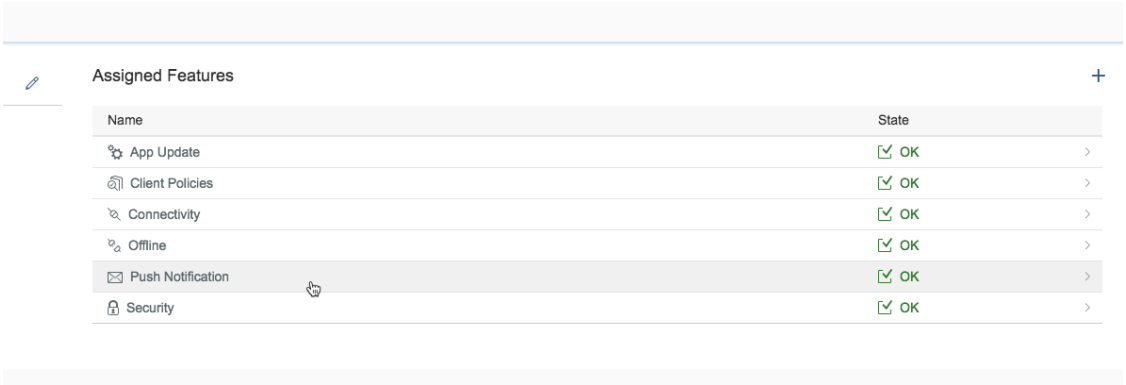
To configure and activate push for the Android platform, see the Configuring Push for Android [page 96] procedure.

Procedure

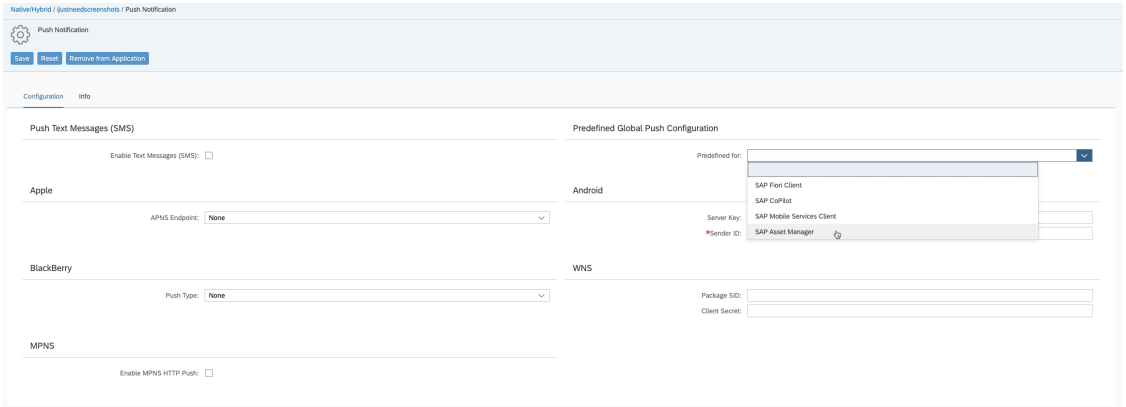
1. **Configure the SAP Cloud Platform mobile services push API:**
   a. Enable the *Push Notification* feature in SCPms:



b. Navigate to the *Push Notification* in the and create a new HTTP connection to your external server named *Assigned Features*:



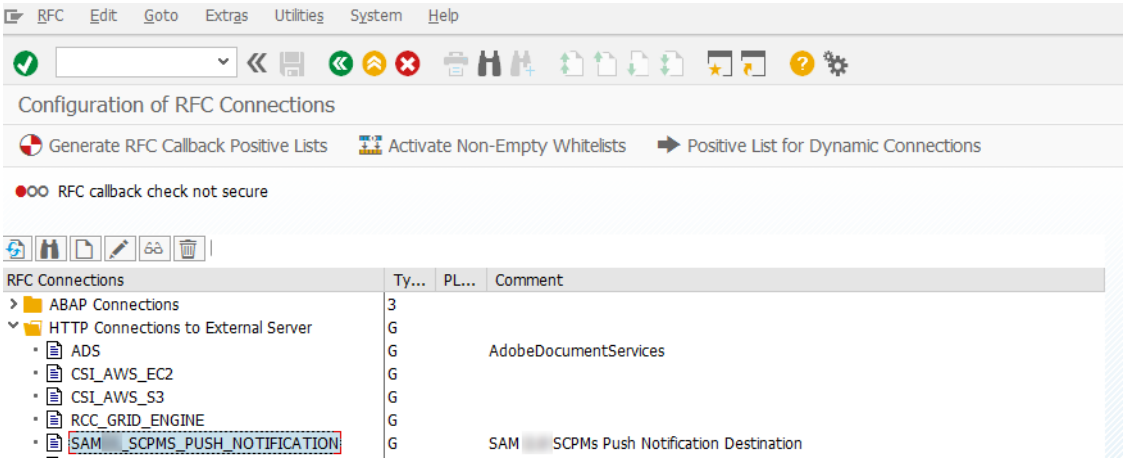
c. If you are using a custom deployment of SAP Asset Manager, upload the corresponding APNs certificates here. If you are using the default application provided by the Apple App or the Google Play store, select *Predefined for* SAP Asset Manager in the *Predefined Global Push Configuration* section.



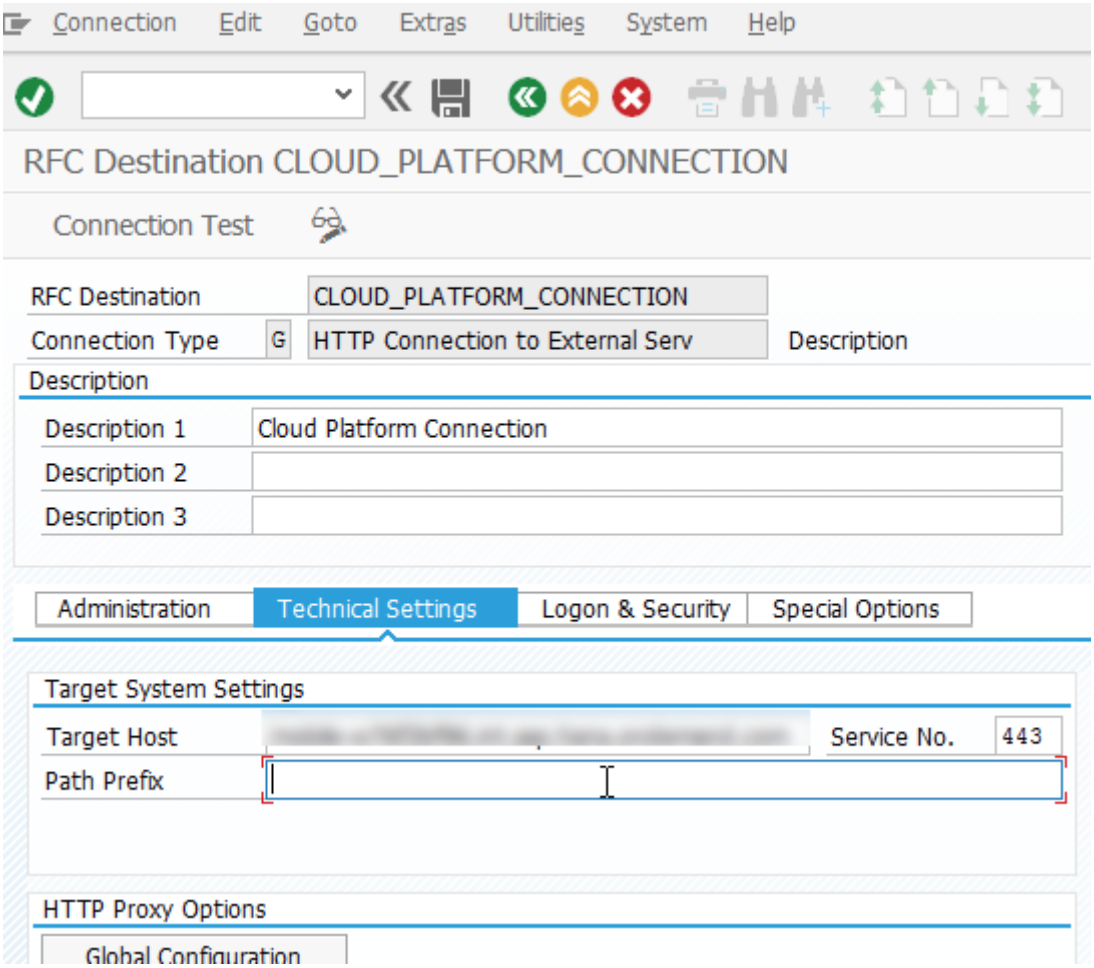
The SAP Cloud Platform mobile services is configured for push.

2. **Configure the back-end system to utilize the SCPms push APIs:**

a. Using the SAP GUI, run transaction *SM59*. Create a new HTTP connection with the name *SAM1911_SCPMS_PUSH_NOTIFICATION*.



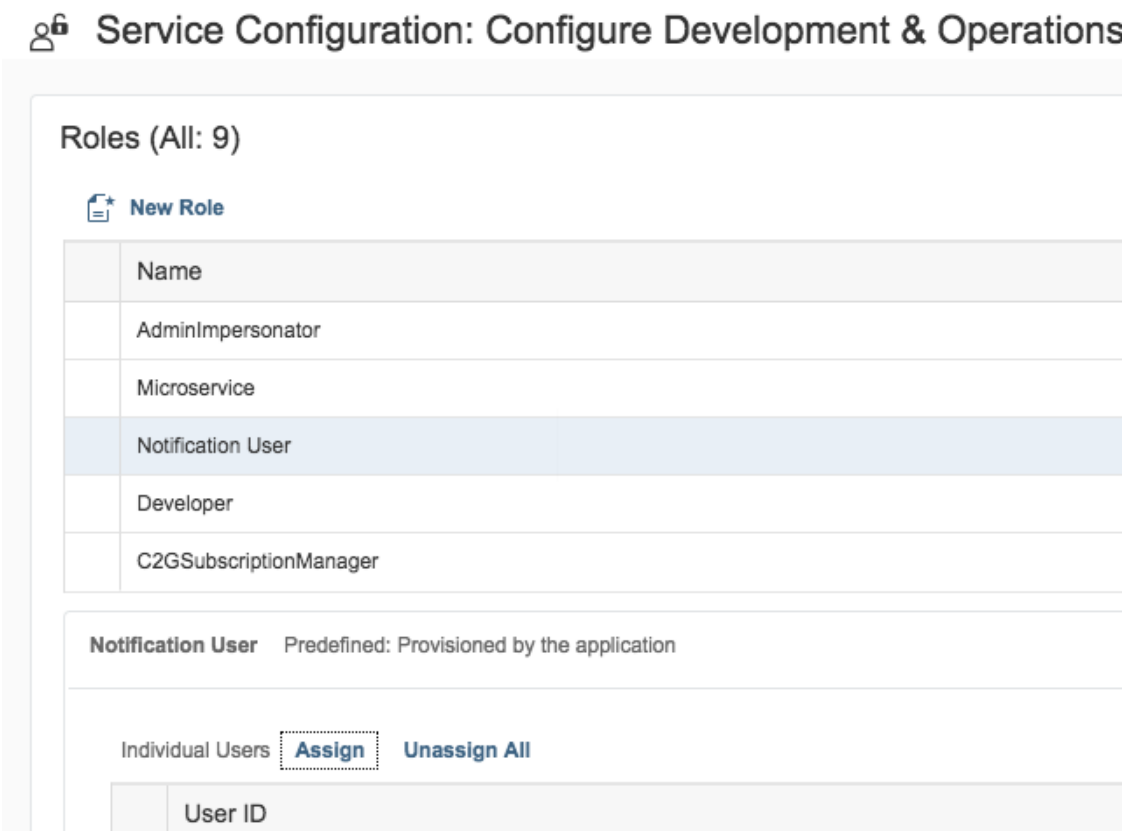
b. In the *Technical Settings* and tab of the new connection, set the *Target Host* to match the push API of the SCPms, using *443* (the port number for HTTPS connections).



> **Note**
>
> If necessary, configure the proxy you are using to allow your back end to connect to the outside internet.

c. Click the *Logon & Security* tab. Under the *Logon Procedure*, select *Basic Administration*. Enter the user name and password of a service user.

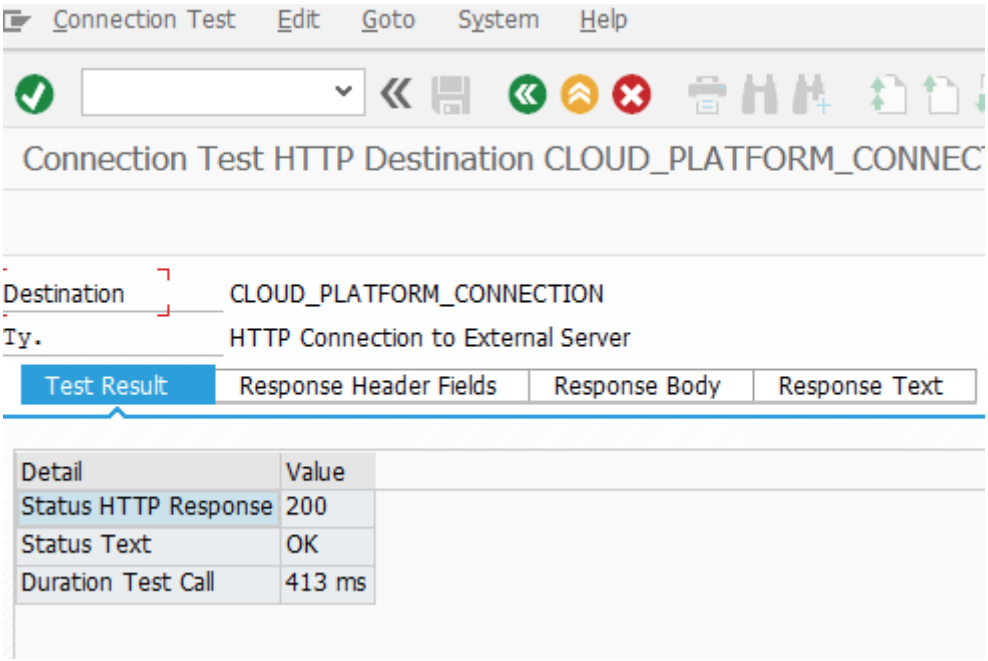
d. In SAP Cloud Platform, ensure your service user has the role of *Notification User* assigned to them to ensure that the user is allowed to utilize the SCPms. The service user must be a member of the SAP Cloud Platform account.



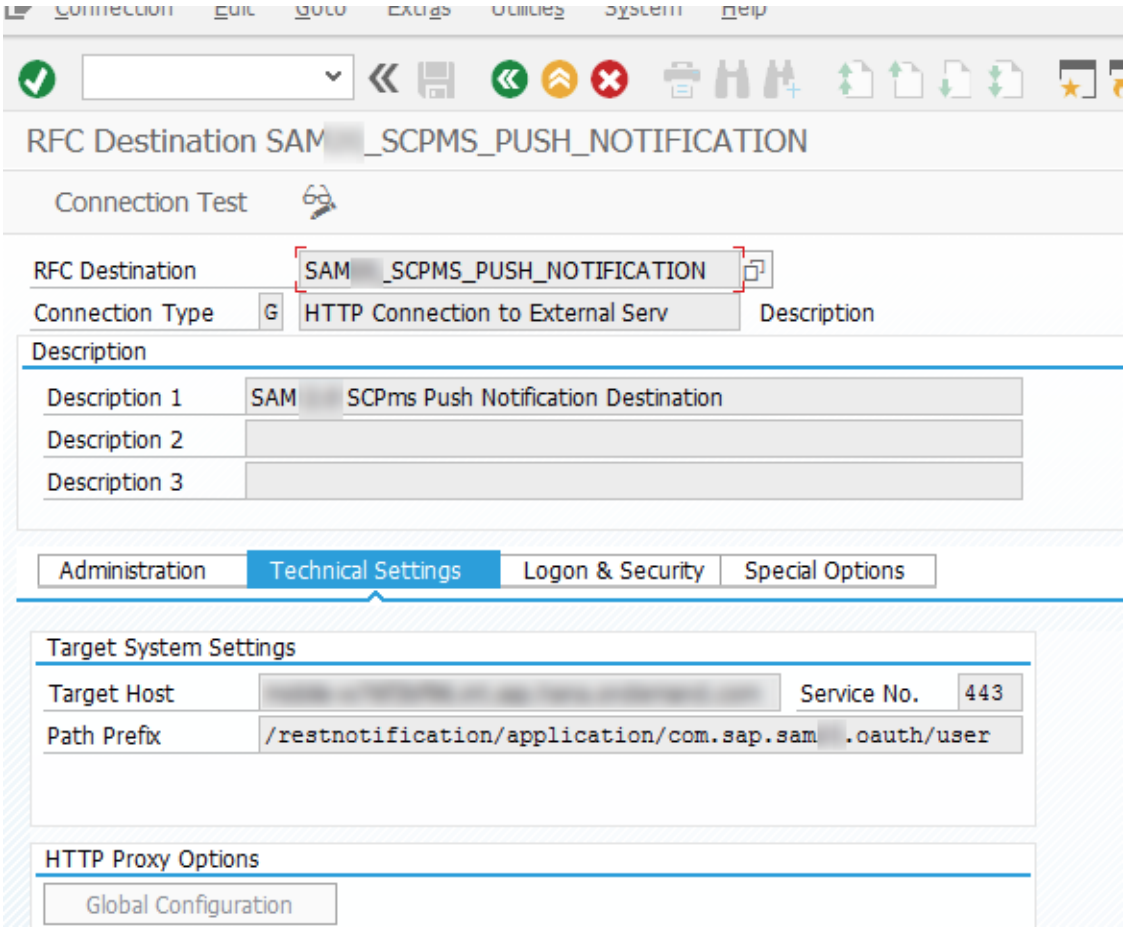
e. In the *Security Options* section of the *Logon & Security* tab, ensure that the `<SSL Secure Protocol>` is set to *Active*.

f. Remaining in the *Security Options* section, ensure that the *SSL Certificate List* used contains the SAP Cloud Platform certificate chain. You can check the certificate chain in the STRUST transaction by looking at the SSL client certificate list. View the SCPms certificate chain in any web browser by opening up the SCPms cockpit and checking the security settings.

g. Save the connection and perform a connection test. If the configuration is completed properly, a 200 HTTP response is returned.



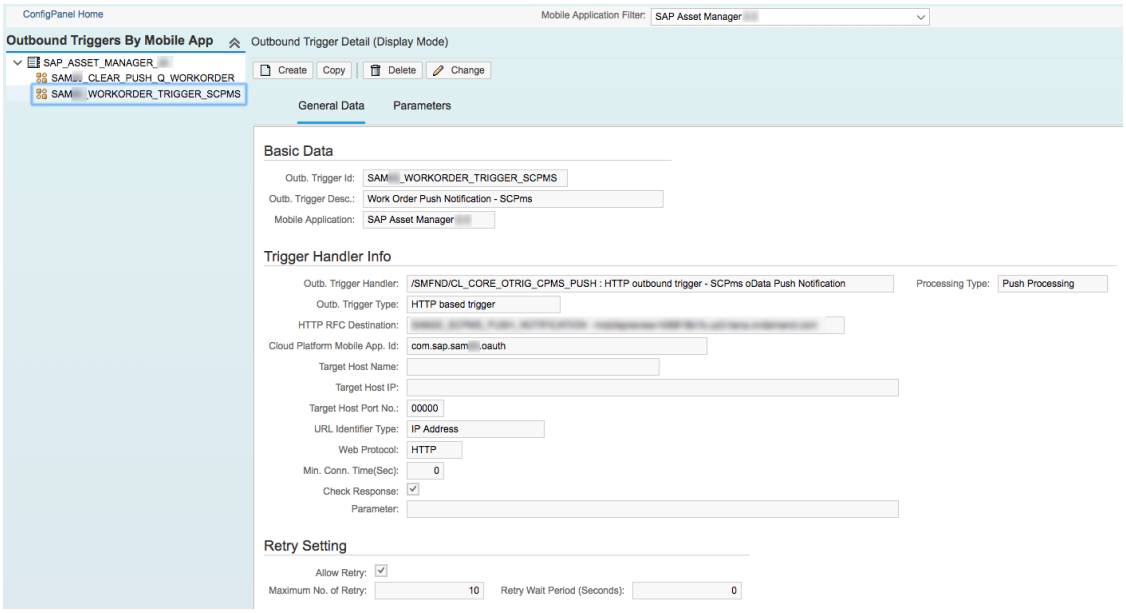
h. If the previous step correctly returned a status of *200*, add the rest of the path for the API to allow the POST to work properly.



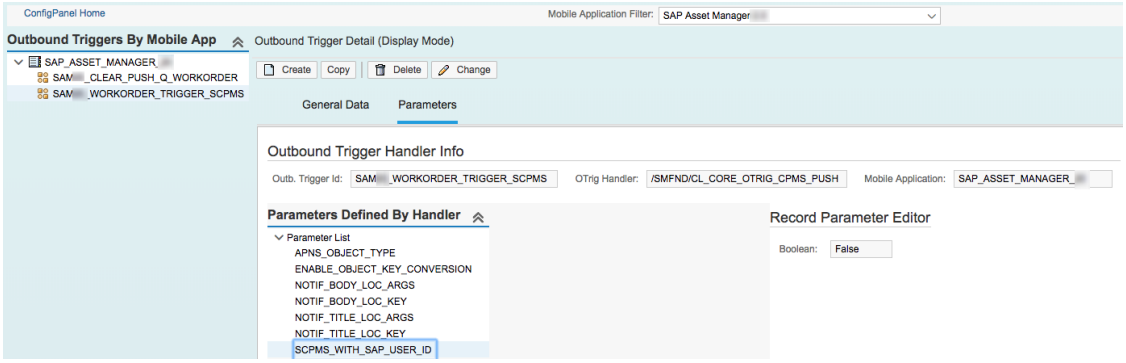
The back-end system can now send push notifications.

3. **Configure the back end to enable push from the SAP Asset Manager application:**

   a. Using the ConfigPanel, navigate to the *Outbound Trigger Configuration* panel, *General Data* tab. Check the outbound triggers to ensure that the previously established *HTTP RFC Destination* is set as the RFC destination to the APPID that provides the push notifications to the mobile devices.



   b. Click the *Parameters* tab. If your user store on the back end and user store on the SAP Cloud Platform are identical, set the *SCPMS_WITH_SAP_USER_ID* parameter value to *True*. If the user stores are not identical, set the parameter to *False*.



   c. Return to the ConfigPanel home screen and click the *Push Scenario Definition* link. Navigate to the *Outbound Trigger* tab. Find and highlight the push on the list of *Push Scenarios by Mobile App* and ensure that the outbound trigger is active.

Results

Push services are activated for SAP Asset Manager. Thoroughly test the push functionality before deploying to the client devices.

3.4.6 Configuring Push for Android

Firebase Cloud Messaging (FCM) is a cross-platform cloud solution for messages and notifications for Android, iOS, and web applications.

Context

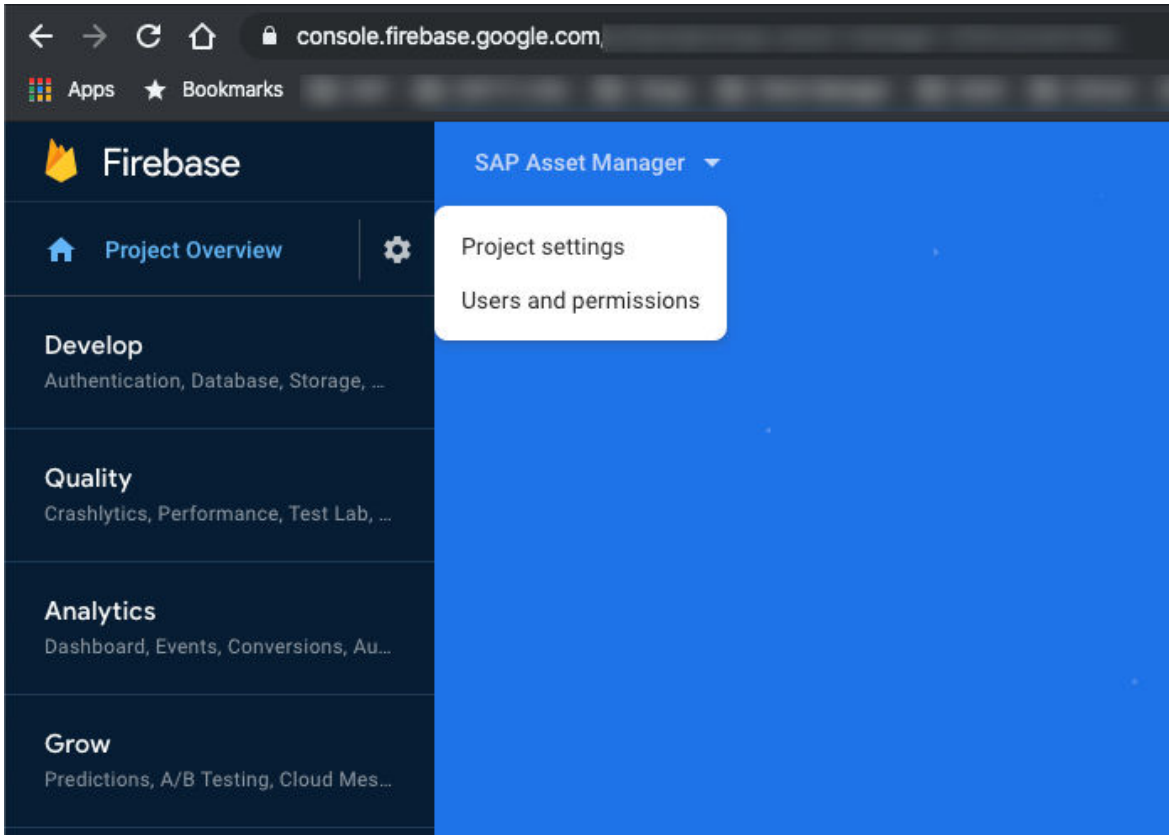
To enable push notification for the SAP Asset Manager application using the Android platform, use the following procedure:

Procedure

1. Create a free Firebase account. See the main Firebase page to set up a new account, or connect an existing account.

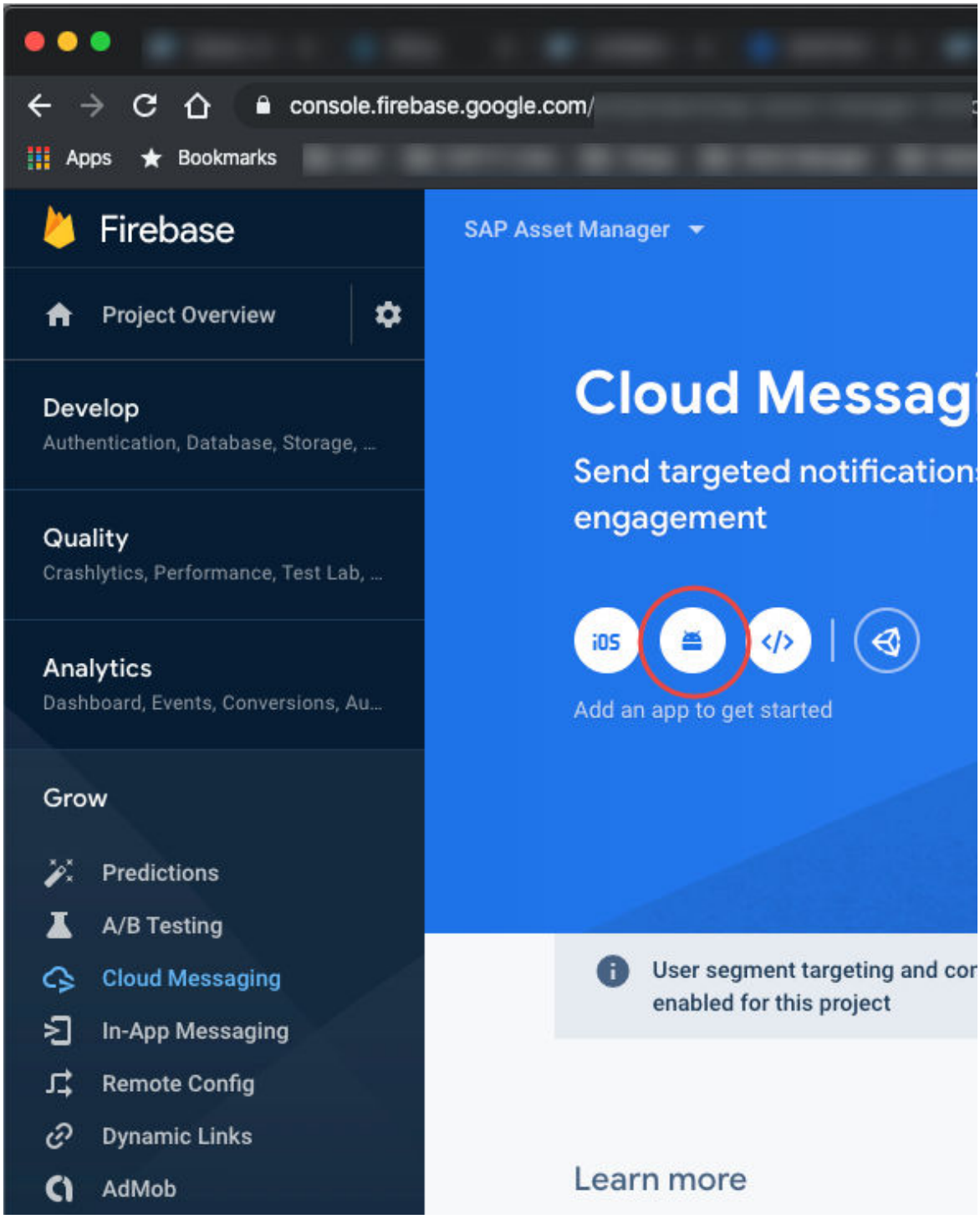
   > **i Note**
   >
   > Documentation on Firebase is found on the Firebase Documentation page.

2. Add a new project, or use an already-existing project.
3. Click the gear icon and select the *Project Settings* menu option. Click the *Cloud Messaging* tab.



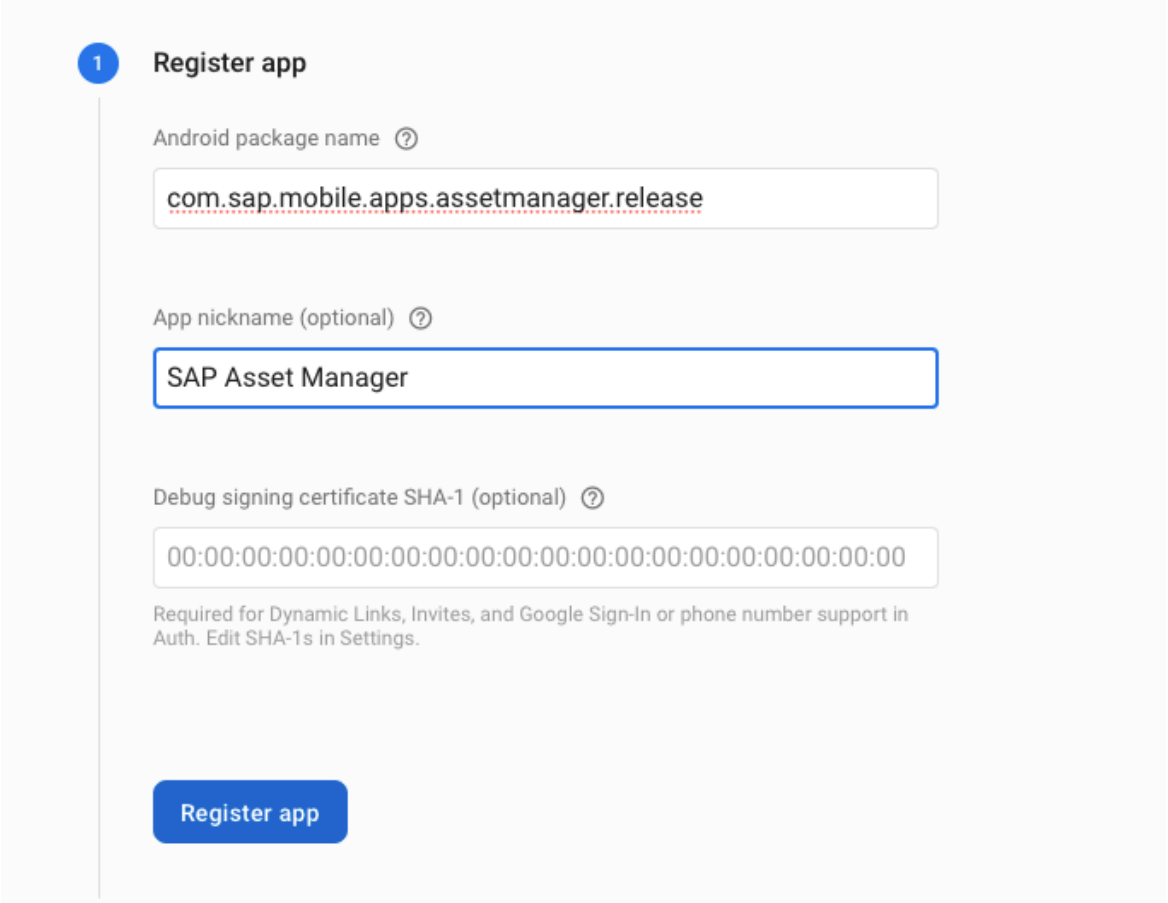
4. Copy and paste the information in the *Server Key* field to use in a later step.
5. Navigate to *Grow* > *Cloud Messaging* using the sidebar.
6. Click the Android icon to add a new Android app.

   The *Add Firebase to your Android app* window appears.



7. Add your package name to the *Android package name* field in *Step 1*. Ensure that the package name matches the *BundleID* found in the `SAM.mdkproject/MDKProject.json` file. When done, click *Register app*.

   Your app is registered and you're moved to *Step 2 - Download config file*.



8. Click the *Download google-services.json* button located in *Step 2 - Download config file*.

   The `google-services.json` file is downloaded to your computer.

9. Navigate to `SAM.mdkproject/App_Resources/Android`. Paste the `google-services.json` file you downloaded in the previous step to this location.
10. Build the SAP Asset Manager client. For detailed instructions on building the client, see SAP Asset Manager Installation Guide, *Building / Branding the SAP Asset Manager Application* chapter.
11. Copy and paste the server key you saved in *Step 3* inside your SCPms push notification configuration in the *Android* section.

    For instructions on how to enable Android push notifications in SCPms, see the Android Push Notifications procedure.

    For detailed information on configuring push for the SAP Asset Manager application, see the procedure Activating Default Push Services for SAP Asset Manager, specifically the screenshot in *Step 1d*.

Next Steps

Continue to the procedure Setting up the Outbound Trigger for your Push Configuration [page 86].

Related Information

https://help.sap.com/viewer/DRAFT/f15c174c3c3647088d38fb220e42c006/latest/en-US/5629175b49304549b724bc838c9a72ec.html

4 Setting up an OData Mobile Data Object

For OData troubleshooting information, see OData API in the *SAP Cloud Platform* documentation.

4.1 Supported OData Features for SAP Mobile Add-On

The following table lists the OData features that SAP Mobile Add-On supports.

OData Feature	Status
Create	Maps to HTTP POST method
Read	Maps to HTTP GET method
Update	Maps to HTTP PUT method
Delete	Maps to HTTP DELETE method
Query	Maps to HTTP GET method
$select	Specifies a subset of properties to return
$top	Determines the maximum number of records to return
$skip	Sets the number of records to skip before it retrieves records in a collection
$filter	Specifies an expression or function that must evaluate to *true* for a record to return to the collection
$count	Returns the number of records in a collection
$orderby	Determines which values are used to order a collection of records
$expand	Specifies that related records must be retrieved in line with the record or collection being retrieved. For example, use *$expand* to retrieve a customer and all orders placed by that customer in a single query.
$inlinecount	Supported
$skiptoken	Supported

OData Feature	Status
$format	Supported
Navigation	Supported
Delta token	Supported
Tombstone	Supported
Complex types	Supported
$batch	Supported
Deep insert	Supported via single post operation and through $batch request using content ID referencing
Custom query options	Not supported
$link	Not supported
$value (media links or attachments)	Supported
ETags / concurrency control	Supported
$filter Details	

String functions
• Supported:
  ○ bool substringof(string p0, string p1)
• Not Supported:
  ○ string trim(string p0)
  ○ string concat(string p0, string p1)
  ○ int length(string p0)
  ○ int indexof(string p0, string p1)
  ○ string replace(string p0, string find, string replace)
  ○ bool endswith(string p0, string p1)
  ○ bool startswith(string p0, string p1)
  ○ string toupper(string p0)
  ○ string substring(string p0, int pos)
  ○ string substring(string p0, int pos, int length)
  ○ string tolower(string p0)
	Partially supported

Date functions	Not supported
Math functions	Not supported
Arithmetic operators	Not supported

OData Feature	Status
Type functions	Not supported
$filter supported on complex type properties within an entity	

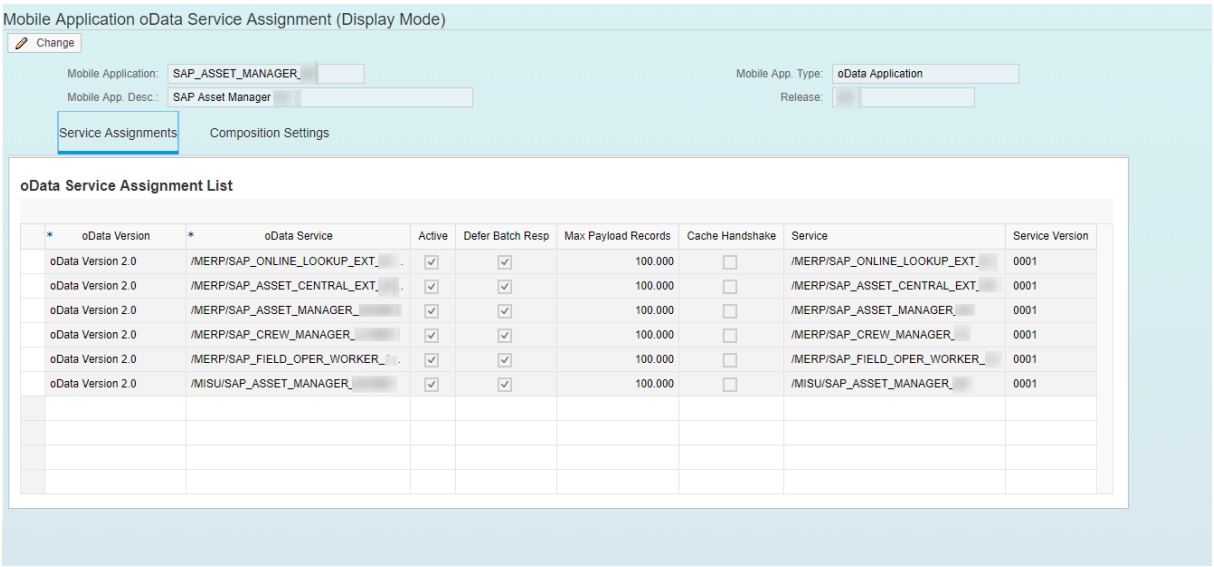
> **Note**
>
> For related constraints, see SAP Note 1830712.

4.2 Setting the OData Mobile Data Object Service Assignment

You can assign SAP system aliases to a service. With the assignment, an OData request from an SAP Gateway consumer can be routed to the corresponding back end service.

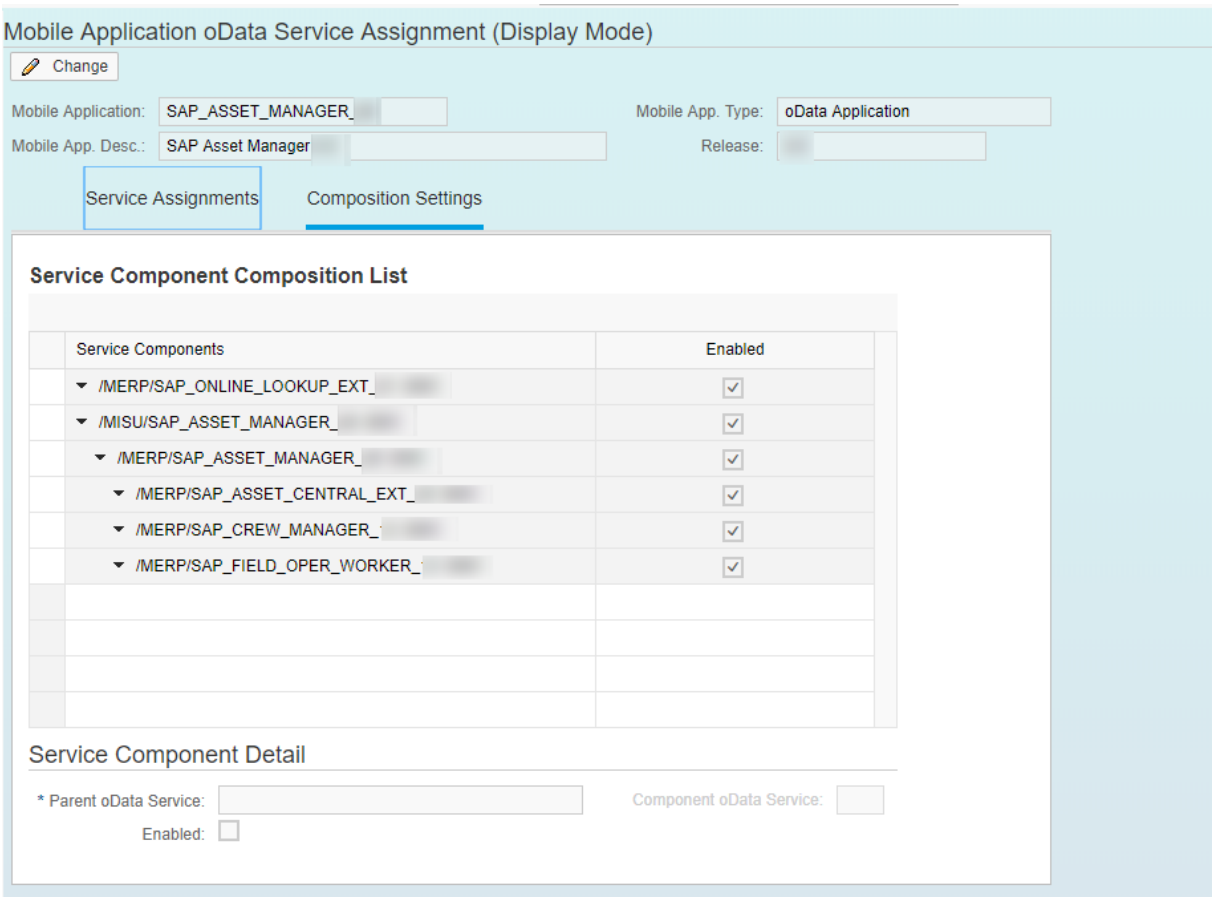
Context

Assign OData services to the SAP Asset Manager application using the *Service Assignments* tab.



Build a hierarchy between assigned services using *Composition Settings*. To utilize OData entities from a different service such as the Crew Management and Field Operations Worker component service, add the

relevant OData services (*/MERP/SAP_CREW_MANAGER_<XX>* and */MERP/SAP_FIELD_OPER_WORKER_<XX>*) under the SAP Asset Manager service as shown in the following example:



Procedure

1. Ensure that your mobile application is selected in the *Mobile Application Filter* field at the top of the page.
2. Expand the *Mobile Application List* in the left pane and select your mobile object.

   Your chosen mobile application OData service assignment details are displayed in the main window on the *Service Assignments* tab.
3. Click the *Change* button to change the existing mobile service assignment details or to add a new mobile service assignment.
4. To add a new mobile service assignment, click the *Assign OData Service* button.
   a. Select an *OData Version*, if there is more than one to choose from, from the dropdown menu.
   a. Select an *OData Service*, or system alias, from the dropdown menu.

      The corresponding back-end server is populated in the *Service* field.
   b. To *Defer Batch Response*, mark the checkbox. That is, you are setting the OData service to process all of the CHANGESET operations at once (deferred processing).
   c. To activate your new service assignment, check the *Active* checkbox.

5. On the *Composition Settings* tab, build a hierarchy between your OData services to include all metadata from different child services. You can add and remove component assignments depending on your service needs.
6. To add a component assignment, choose *Add Component Assignment*.
    a. Select a *Parent OData Service* from the dropdown list.
    b. Select *Component OData Service* from the available list to make it a child service for the parent OData service.
    c. To activate the new component assignment, check the *Enabled* checkbox.
7. To remove the component assignment, select the service component you want to remove and choose *Remove Component Assignment*.

    The hierarchy relationship with its parent service component is removed.

Next Steps

Perform the following procedures:

- Setting the OData Mobile Data Object Configuration [page 104]
- Setting the OData Model Configuration [page 106]

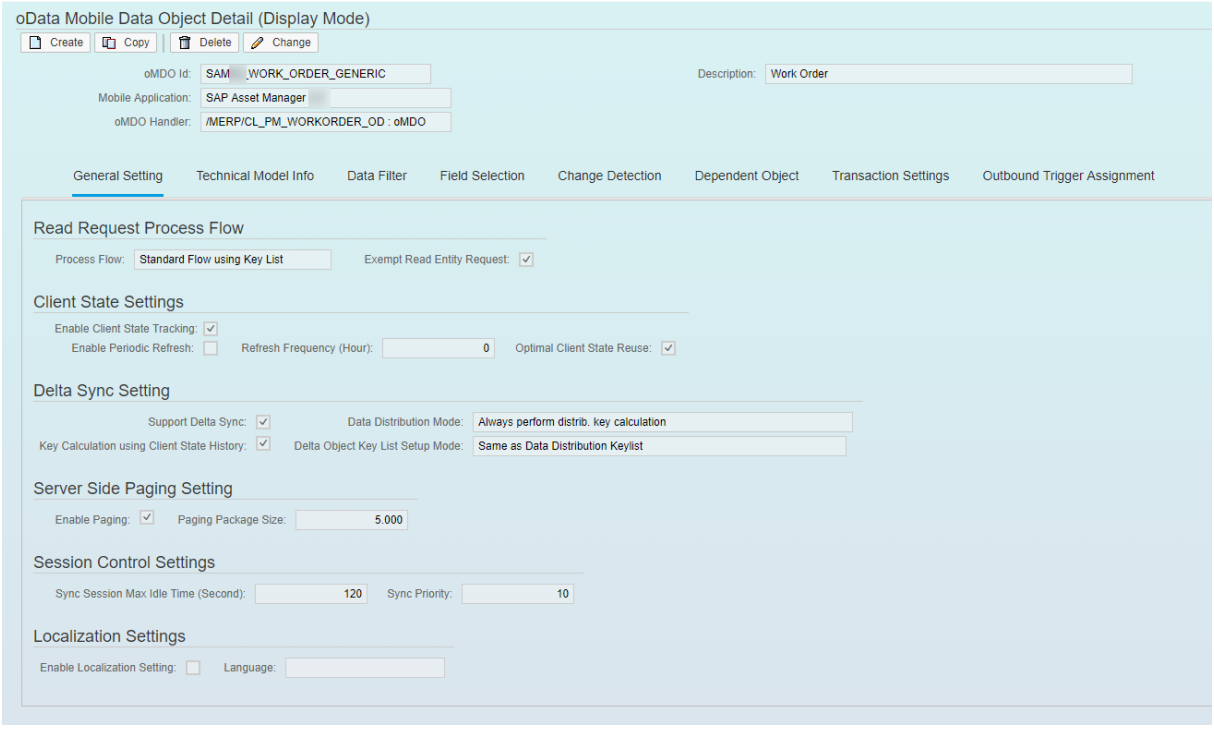
4.3 Setting the OData Mobile Data Object Configuration

Prerequisites

If you are setting up a new OData mobile data object, or changing an OMDO, read and perform the following procedures before performing this procedure:

- Setting the OData Mobile Data Object Service Assignment [page 102]

Context



Procedure

1. Navigate to and click the *Mobile Application Integration Framework Configuration Home* > *OData Mobile Data Object Configuration* link.

   The *OData Mobile Data Object Configuration Detail* page displays.
2. Select your application in the *Mobile Application Filter* field. Selecting an application filters the OData Mobile Data Object by Mobile App choices in the left panel to only those available in your chosen application.
3. If you are copying an OMDO, which is recommended, choose your OMDO from the *OData Mobile Data Object by Mobile App* list on the left and copy it to your customer namespace. See the Copying an Object to the Customer Namespace [page 70] procedure for more details. Do not modify the original OMDO.
4. Enter an OMDO ID in the field with customer namespace.
5. Select a *Mobile Application* from the dropdown list.
6. Select an *OMDO Handler* from the dropdown menu. The OMDO handler is the name of the handler as defined in the system.

   The OMDO handler will provide the data source for the entity record.
7. Enter a short *Description* of your new OData mobile data object.
8. Choose one of two settings for the *Process Flow* in the *Read Request Process Flow* section:
   - **Standard Flow Using Key List**

- **Basic Flow without Key List**

9. You can keep the *Client State Settings* from the original OMDO, or change it depending on your business needs. and from the original OMDO, or change it depending on your business needs. and *Delta Sync Settings* from the original OMDO, or change it depending on your business needs. and from the original OMDO, or change
10. Check the *Enable Paging* checkbox to set the package size for the SCPms. Type in a number for the default package size in the *Paging Package Size* field.

Next Steps

Continue to Setting the OData Model Configuration [page 106].

4.4 Setting the OData Model Configuration

An OData model gives detailed information about each object in an OData feed. You can define a new data model in your application to suit your requirements based on the data you want expose at runtime.

Prerequisites

Complete the following procedures:

- Setting the OData Mobile Data Object Service Assignment [page 102]
- Setting the OData Mobile Data Object Configuration [page 104]

Context

Entity Sets are used to group instances of an entity type together with instances of any type that are derived from this particular entity type. You can access the OData entity details from the ConfigPanel home page by choosing *OData Model Configuration*.

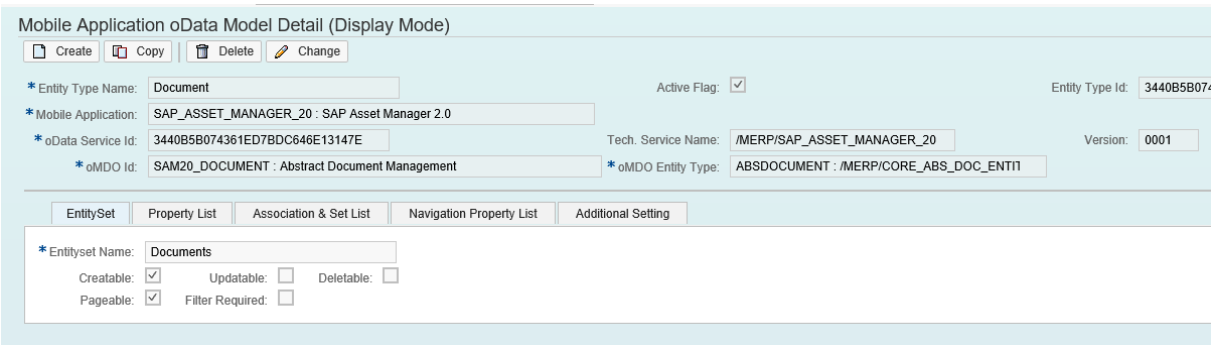
You can define properties for entity types on the **Property List** tab. Properties define the characteristics of data that an entity type instance contains at runtime.

An association is a named relationship between different entities. An association defines a peer-to-peer relationship between participating entity types, and it can also support different multiplicities at both ends. You can define associations between current and dependent entities on the **Association & Set List** tab. An association set groups association instances, and specifies the relationship between two entity sets based on the respective entity types of the underlying association.

Navigation properties describe the association relationship between two entities. The navigation property is tied to an association, and it allows the navigation from one end of the entity type, which declares the

navigation property, to the other related end. The entity types include one or more navigation properties and can be defined on the *Navigation Property List* tab.

Finally, you can set the bind structure conversion exits and the Media flag for entity type on the *Additional Setting* tab.



> **i Note**
>
> Optional steps are included to explain the required fields when creating a new OData model. These fields are grayed out when you are working with a copied OData model and you can ignore them in the procedure.

Use the following steps as a guide to change an existing OData model:

Procedure

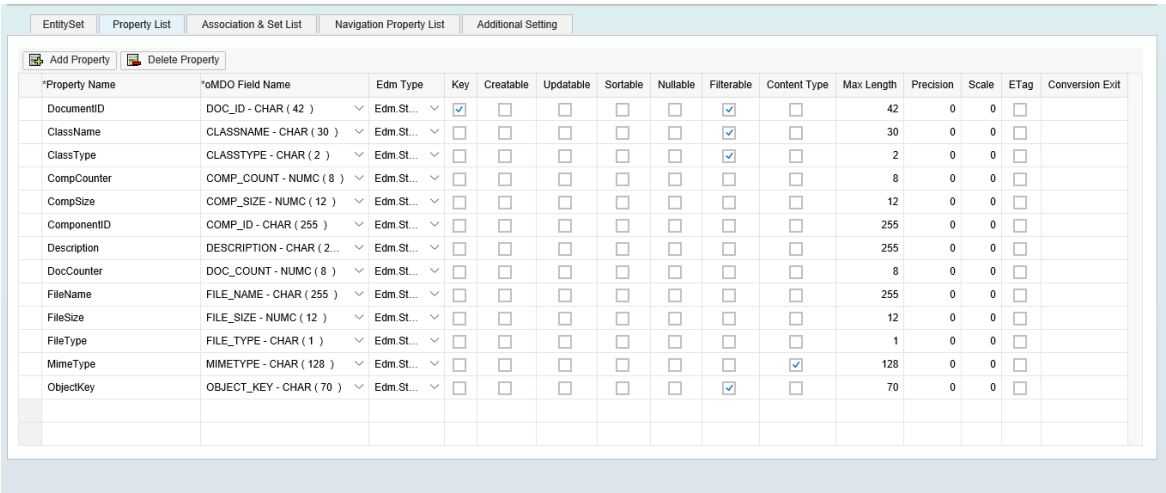
1. Navigate to and click the *Mobile Application Integration Framework Configuration* > *OData Model Configuration* link.

   The *Mobile Application OData Model Detail* page displays.
2. Select your application in the *Mobile Application Filter* field. Selecting an application filters the OData Model Entity Type by Mobile App choices in the left panel to only those available in your chosen application.
3. Once you have selected the mobile application, select *OData Service ID* from the drop-down list.

   Note that you cannot share models between OData services. Each service has its own model.
4. If you are creating a new OData model, click on *Create* button on the top and type an entity type name in the field. The entity type name represents the structure or a single record.
5. Select an *OMDO ID* from the drop-down list. The OMDO ID is the object that is providing the data for the record.
6. Select an *OMDO Entity Type* from the drop-down list. The OMDO entity type is the source that provides information to the OData model. When a service request for the entity type occurs, the OData model invokes the selected OMDO ID and the related handler method.
7. Type an *EntitySet Name* into the field. While an entity type describes a data structure, an entity set contains the instances of the given structure. Therefore, a best practice for an entityset name is to create a plural of an entity type name. For example, if an entity type name is *Test*, the entityset name will be *Tests*.
8. Check any of the following checkboxes to enable additional OData features. Note that some may require additional configuration on other tabs or links.
   - *Createable*: Similar to a POST request in REST

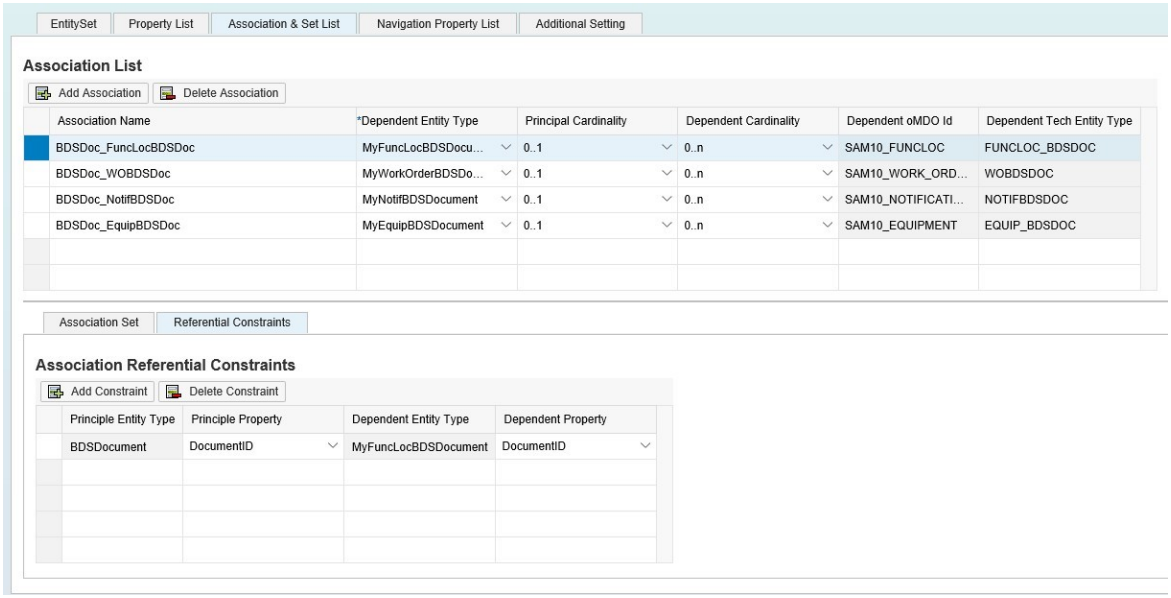
- *Pageable*: Enables data staging. See Setting the OData Mobile Data Object Configuration [page 104], *General Setting* tab, *Server Side Paging Setting* section, for more information.
- *Updateable*: Similar to a PUT or MERGE request in REST
- *Filter Required*: When checked, the entityset cannot be accessed directly. That is, the entityset is only valid within its containing entity and is not visible in the service document. You cannot query the entityset without using a `$filter` expression. If you try to access the entity set without using a `$filter` expression, a message displays that explains which filter expressions are required, at a minimum.
- *Deletable*: Similar to a DELETE request in REST

9. Click the *Property List* tab.



10. To add a new property to the entity type, click the *Add* button.
    a. Type the property name into the `<Property Name>` field.
    b. Select an *oMDO Field Name* from the dropdown list.
    c. Select the appropriate *EDM Type* (Entity Data Model) from the dropdown list.
    d. Check the *Key* column for Key fields.
    e. Define the attributes of the new property depending on the scope of the entity type.

    If you use the *Datetime* Edm Type and its related properties as an optional field, set the attribute *Nullable* to true.
11. Click the *Association & Set List* tab.

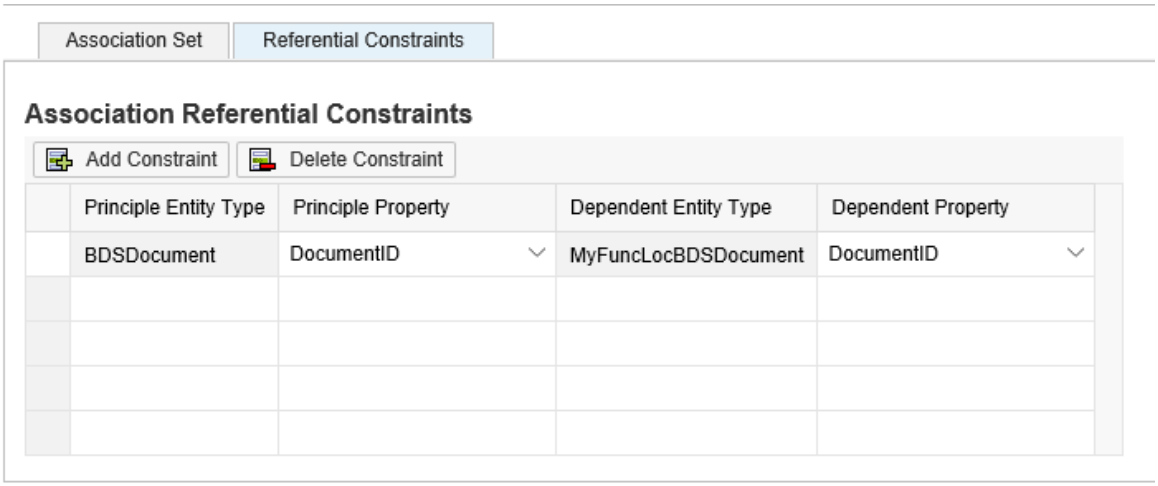


Define associations between entities to express relationships between entities.

Associations themselves are freestanding. Specify on top of the associations, which of the entities participating in the relationship can navigate over the association to the other entity using the *Referential Constraints* tab.

12. Click the *Add Association* button to add a new association. Associations define a peer-to-peer relationship between participating entity types, and can support different multiplicities at both ends.
    a. Type a name for your new association in the *Association Name* field.

       Your Association can be either internal or external when adding a new association; by default the current entity will be the principle entity. If you want to add an external association where the current entity is treated as dependent entity, select the *External Association* checkbox.
    b. Select the dependent entity from the *Dependent Entity Type* drop-down menu for internal association, whereas select the *Principle Entity Type Id* from the drop-down for external association.
    c. Choose the *Principle Cardinality* and the *Dependent Cardinality*. Both use the following cardinality rules. Note that many-to-many relations are not supported in SAP Asset Manager
       - 0..1: Only one instance occurs; zero is also allowed
       - 1: One-to-one relations. Exactly one instance occurs
       - 0..n: Zero-to-many relations. Zero or more instances occur
       - 1..n: One-to-many relations. One or more instances occur
    d. Select the *Principle/Dependent OnDelete Cascade* checkbox, if you want to delete an associated collection when a principle or related parent entity got deleted from the mobile device. This feature only works with local objects.
    e. Type the name of your association set in the *Association Set Name* field under *Association Set*.
13. Click the *Referential Constraints* tab to add or change a referential constraint.

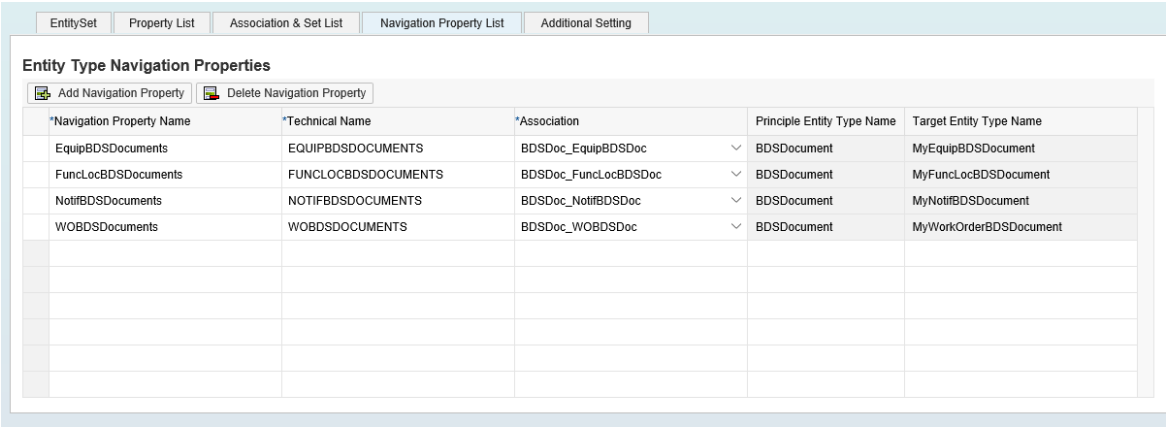


Referential constraints ensure that the principle entity that is referenced always exists. You can use a referential constraint to specify the foreign key relationship for a 1:1 relationship or a 1:n relationship.

You have to match the key properties of the principle entity type with the properties from the dependent entity type that correlates to the key property of the principle type. Populate all key properties from the principle entity type.

a. Click the *Add Constraint* button to add a new referential constraint.
b. Select the *Principle Property* and the *Dependent Property* from the drop-down menus. As mentioned earlier, the entities must be identical and all key properties must be populated from the principle entity.

14. Click the *Navigation Property List* tab to create a navigation property for entity types.



The navigation property is tied to an association, and it allows the navigation from one end of the entity type that declares the navigation property to the other related end.

> **i Note**
>
> If you add a new navigation entity, first add a new association for it through the *Association & Set List*. Set the association cardinality for both principle and dependent entities.

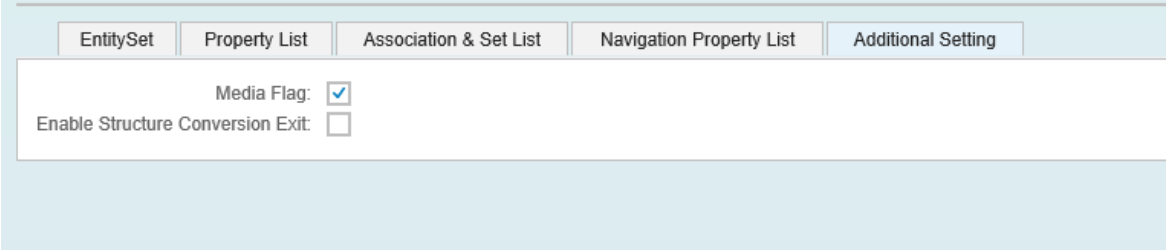
15. Click the *Add Navigation Property* to add a new navigation property.

You can create a navigation property for both principle and dependent entity type using the same association so that link will be created in both directions.

a. Type a name into the *Navigation Property Name* field.
b. Type a technical name into the *Technical Name* field.
c. Choose an *Association* (coming from the *Association & Set List* tab) from the drop-down menu.

   The *Dependent OMDO ID* and *Dependent Tech Entity Type* cells are populated based on which association entity you choose.
d. Repeat these substeps to create the navigation property on the remaining principle or dependent object.

16. Click the *Additional Setting* tab.



a. Select the *Media Flag* checkbox for media-related entity types to trigger the download of media content on the entity set collection.
b. Select the *Enable Structure Conversion Exit* checkbox to allow the SAP Asset Manager application to access the OData channel. The OData channel delegates handling of conversion exits, currency, currency amounts, units of measurement, and unit amount conversions to the SAP Gateway framework.

Results

Once the model is fully defined, when a client makes an HTTP request, it is calling for the metadata for an OData service. The SAP Gateway returns an XML string to the client, which is also reflected in the ConfigPanel.

5 Meter Management Configuration

The Meter Management component is delivered out of the box with predefined settings, which you can change according to your back-end system setup. The following settings, however, have to be set:

- Binding Industry Solutions & Utilities (ISU) process type to work order type
- Setting the optimal meter reading history
- Binding meter reading reason relevant for technical installation
- Binding meter reading notes based on the ISU process type

5.1 Binding ISU Process Type to Work Order Type

Binding the ISU process type to the work order type provides the SAP Asset Manager application the correct representation of what process type is being conducted with the different work order types. This binding is located in the **SAM1911_ORDER_ISULINK** OMDO, under the *Read* filter. You can update these filters according to your business process.

Assignment types *1*, *2*, and *3* are supported for the Meter Management component.

The following filters represent binding criteria for different process types. By default, the ISU process types are bound to order types as follows:

Filter Name	Type	Value	Comments
ORDTYPE_DISCONNECT	Standard Filter Mandatory	DC01 RC01	Used for the disconnect process
ORDTYPE_INSTALL	Standard Filter Mandatory	SM01	Used for the installation process
ORDTYPE_READING	Standard Filter Mandatory	MR01	Used for the meter reading process
ORDTYPE_REMOVE	Standard Filter Mandatory	CU01	Used for the remove process
ORDTYPE_REPAIR	Standard Filter Mandatory	SM02	Used for the repair process
ORDTYPE_REPLACE	Standard Filter Mandatory	RP01	Used for the replace process

To change the default binding for a particular process, complete the steps below:

1. On the ConfigPanel home page, choose *OData Mobile Data Object Configuration*.
   Make sure that you select your desired mobile application in the *Mobile Application Filter* field at the top of the page.
2. From the *OData Mobile Data Object List* select desired OMDO object, such as **SAM1911_ORDER_ISULINK**, and then click the *Data Filter* tab.
3. Expand the *Defined Filters* list under the *READ* operation with the standard filter. Select the filter that you want to update from the list of available filters as listed in the table in this topic. Choose the *Change* button from the menu.
4. Set the order type for the desired process type you have selected.
5. Save your changes.

5.2 Binding Meter Reading Notes Based on ISU Process Type

The back end ISU system configuration specifies which meter reading notes are relevant for the major ISU process types. This configuration has to be replicated in the OData Mobile Data Object **SAM1911_METER_READING_NOTE** under the *READ* operation with the standard filters, so that the SAP Asset Manager application reflects the proper meter reading notes for a specific process type. The default configuration lists the meter reading notes relevant for the ISU process. However, you can change it if you have different requirements in your back end configuration.

Filter Name	Type	Value	Comments
DISCONNECT_NOTE_REMOVE	Standard Filter, Mandatory	01 04 05	Used for the disconnect notes for the remove process
METERREAD_NOTE _INSTALL	Standard Filter, Mandatory	01 04	Used for the meter reading notes for the installation process
METERREAD_NOTE _REMOVE	Standard Filter, Mandatory	04 05	Used for the meter reading notes for the remove process

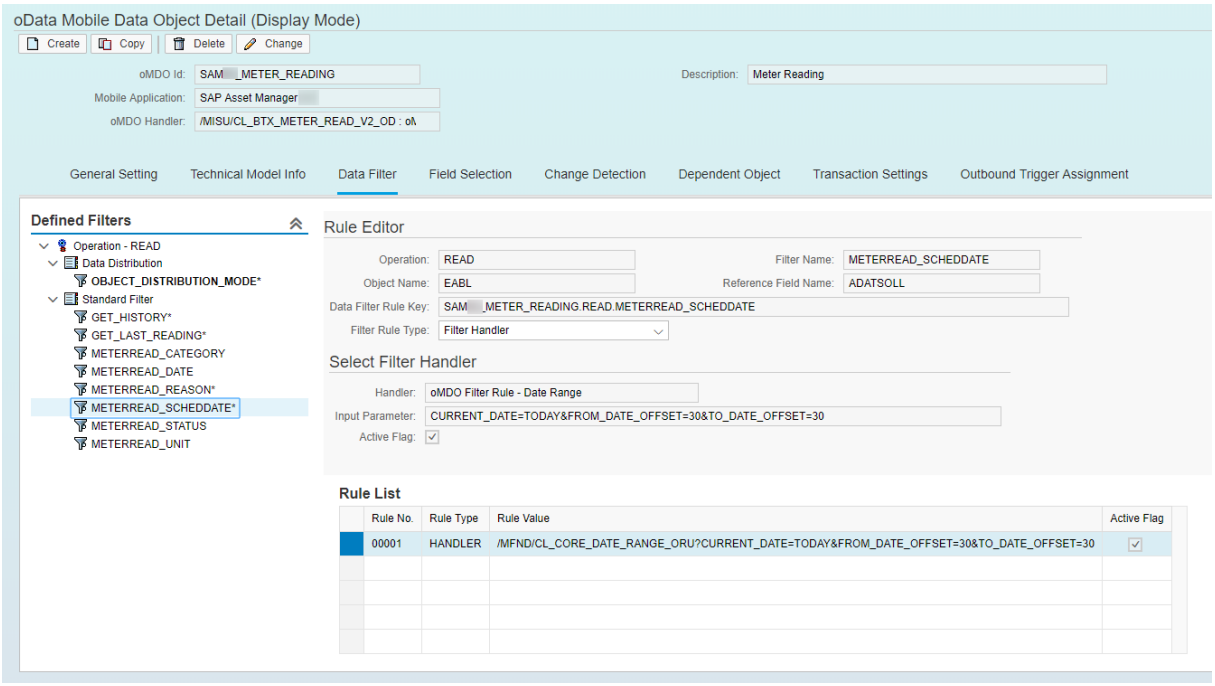
To change the default binding for a particular process, complete the following steps:

1. On the ConfigPanel home page, choose *OData Mobile Data Object Configuration*.
   Make sure you select your desired mobile application in the *Mobile Application Filter* field at the top of the page.
2. From the *OData Mobile Data Object List* select desired OMDO object, such as **SAM1911_METER_READING_NOTE**, and then click on the *Data Filter* tab.
3. Expand the *Defined Filters* list under the *READ* operation with the standard filter. Select the filter that you want to update from the list of available filters as listed above. Choose the *Change* button from the menu.

4. Set the meter reading note for the desired ISU process type you have selected.
5. *Save* your changes.

5.3 Meter Reading History Interval Definition

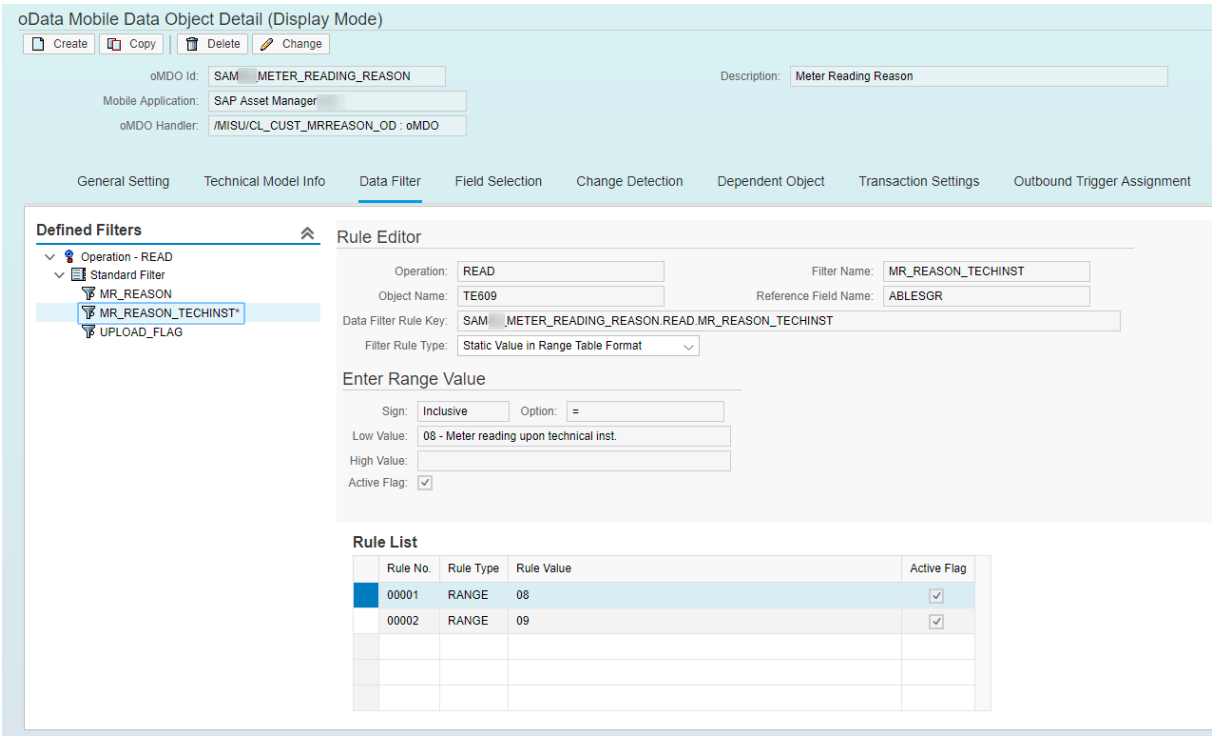
The default setting for meter reading history is to include all meter readings from the past 30 days till the current day. If you have a different requirement, you can change it from the OData Mobile Data Object **SAM1911_METER_READING** under the READ operation with the standard filter *METERREAD_SCHEDDATE* as shown in the following example.



5.4 Binding Meter Reading Reasons Relevant for Technical Installation

The back end ISU system configuration specifies explicitly which meter reading reasons are relevant for technical installation. These are the only reasons displayed on the SAP Asset Manager application when completing a meter reading during the technical installation process. Set this binding in the OData Mobile Data Object **SAM1911_METER_READING_REASON** under the *READ* operation with the standard filter *MR_REASON_TECHINST*.

The default configuration contains meter reading reasons *08* and *09* as required for technical installation. If you have different requirements for your back end configuration, you can change these defaults.


oData Mobile Data Object Detail (Display Mode)
Create
Copy
Delete
Change
oMDO Id:
SAM METER_READING_REASON
Description:
Meter Reading Reason
Mobile Application:
SAP Asset Manager
oMDO Handler:
/MISU/CL_CUST_MRREASON_OD : oMDO
General Setting
Technical Model Info
Data Filter
Field Selection
Change Detection
Dependent Object
Transaction Settings
Outbound Trigger Assignment
Defined Filters
Operation - READ
Standard Filter
MR_REASON
MR_REASON_TECHINST*
UPLOAD_FLAG
Rule Editor
Operation:
READ
Filter Name:
MR_REASON_TECHINST
Object Name:
TE609
Reference Field Name:
ABLESGR
Data Filter Rule Key:
SAM METER_READING_REASON.READ.MR_REASON_TECHINST
Filter Rule Type:
Static Value in Range Table Format
Enter Range Value
Sign:
Inclusive
Option:
=
Low Value:
08 - Meter reading upon technical inst.
High Value:
Active Flag:
Rule List
Rule No.
Rule Type
Rule Value
Active Flag
00001
RANGE
08
00002
RANGE
09

6 Field Operations Worker Configuration

SAP Asset Manager for Field Operations Worker uses the digital core with SAP S/4HANA for task driven activities and rounds. It supports workers who perform asset inspections and checks with focus on measurement points and on smaller services and repairs.

Field Operations Worker, or FOW, is an add-on component to SAP Asset Manager. If you don't see FOW features while using the SAP Asset Manager application, or in the ConfigPanel, your site hasn't installed the component.

Field Operations Worker adds the following functionality to the core SAP Asset Manager application:

- **View routes data:** A route is comparable to a work order in the base SAP Asset Manager application.
- **View stops data:** A stop in Field Operations Worker is comparable to an operation in the base SAP Asset Manager application. A route is composed of one or more stops.
- **View asset information:** An asset in Field Operations Worker is comparable to a piece of equipment in the base SAP Asset Manager application. Assets are located at an FOW stop.
- Use field data capture to take readings on measurement points. Measurement points are located on assets or a set of assets at a route stop.

6.1 Differentiating a Base SAP Asset Manager Work Order from a Field Operations Worker Order

Creating rules based on order types affects synchronization processing and order downloads to the mobile devices of your users who use the Field Operations Worker component.

Prerequisites

Address the following before performing the procedure:

- The order types for work orders that are downloaded to technicians using the Field Operations Worker component are already determined.
- The person performing the procedure has access to the Config Panel and permissions to change settings.

Context

> **i** Note
>
> Field Operations Worker orders are a subset of the base SAP Asset Manager application work orders.

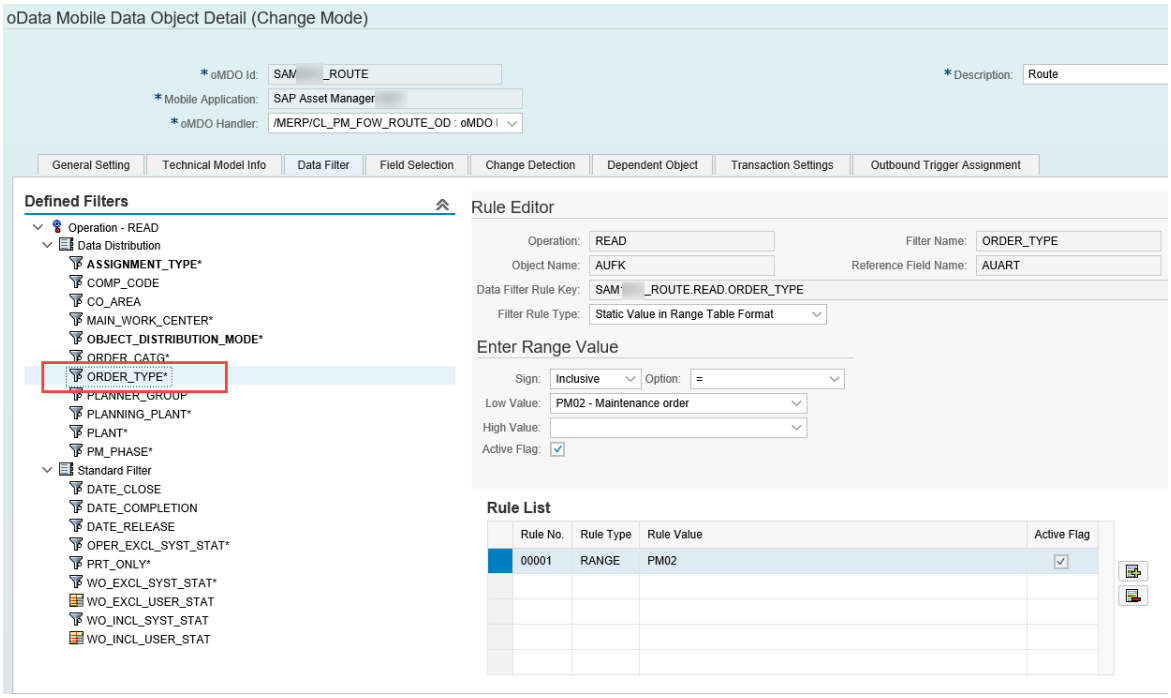
The following procedure modifies the synchronizing behavior of the SAP Asset Manager application, along with the Field Operations Worker component. After you complete the procedure, only orders with a given order type of *PM02* are downloaded to the FOW component. In the procedure, you change the *ORDER_TYPE* filter in the OMDOs involved in order synchronization. Specifically, you add a rule to the filter in the **SAM1911_ROUTE** OMDO to include only the desired order type.

If you don't create a rule for the *PM02* order type, then that order type is excluded from work order download synchronization processing. If the FOW orders are excluded from synchronization processing, then the orders aren't present on the mobile clients of your users.

Procedure

1. Click the *OData Mobile Data Object Configuration* link, then click the *Data Filter* tab from the main ConfigPanel page. Be sure to have your desired mobile application chosen in the *Mobile Application Filter* field at the top of the page.

   Selecting an application filters the *OData Mobile Data Object by Mobile App* choices in the left panel with only OMDOs available in your application.
2. Expand the *OData Mobile Data Object by Mobile App* list on the left and click **SAM1911_ROUTE**.
3. Select the *Data Filter* tab.
4. Click the *Operation - READ* > *Data Distribution* > *ORDER_TYPE* node in the *Defined Filters* list.
5. Create a rule using the following parameters if the rule doesn't already exist:
   - **DOF Rule Type:** Static Value in Range Format
   - **Sign:** Inclusive
   - **Option:** =
   - **Low Value:** PM02 - Maintenance order
   - Ensure the *Active Flag* box is checked



6. Influence Field Operations Worker orders using the following filters found in *Operation - READ* > *Standard Filter* if needed:
    - WO_EXCL_SYST_STAT
    - WO_EXCL_USER_STAT
    - WO_INCL_SYST_STAT
    - WO_INCL_USER_STAT
7. *Save* your changes.

Results

After you finish the procedure, both Field Operations Worker orders and base SAP Asset Manager work orders are downloaded by the SAP Asset Manager application.

7 Configuring SAP Asset Manager to the Asset Intelligence Network

7.1 Configuring Checklist Parameters

Use parameters to enable the checklist feature and configure other checklist options available.

Context

To configure the checklist feature for SAP Asset Manager, use the *CHECKLISTS* parameter group and the following parameters within the group:

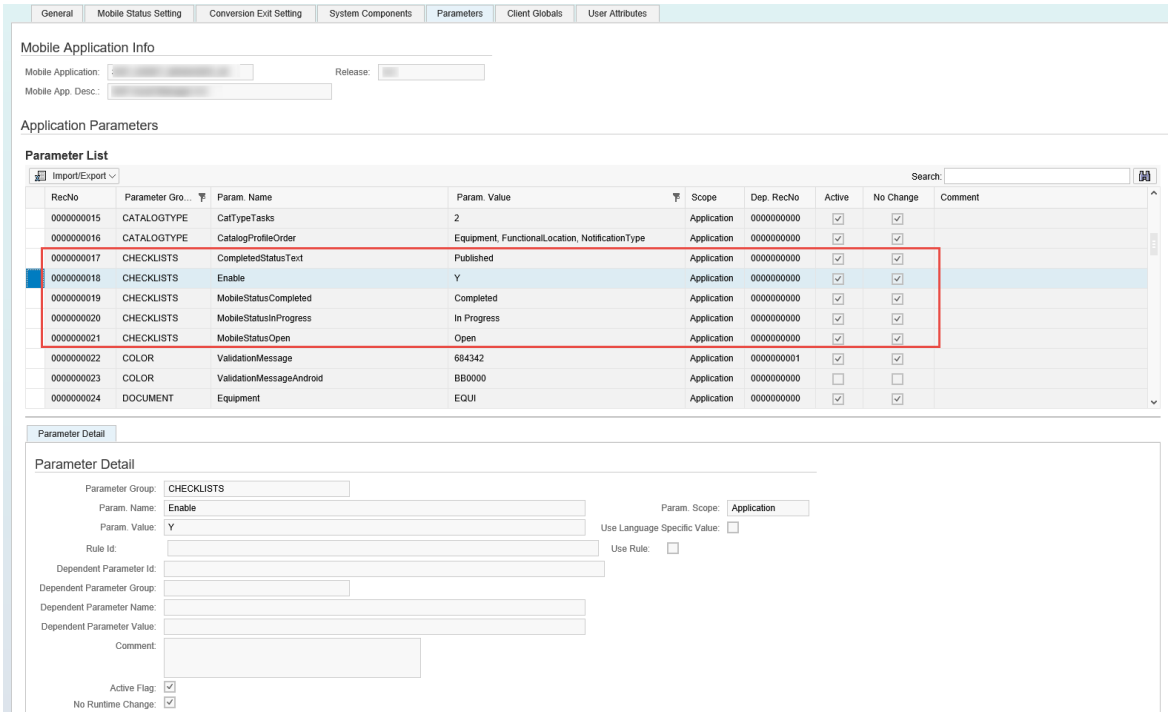
- **Enable:** Out of box is *N*. Set to *Y* to enable the checklist feature.
- **MobileStatusCompleted:** Default is *Completed*. Do not change this setting unless you are integrating SAP Asset Manager with another product besides ASPM.
- **MobileStatusInProgress:** Default is *In Progress*
- **MobileStatusOpen:** Default is *Open*
- **CompletedStatusText:** Default is *Published*. This parameter is used to distinguish completed checklists that have been downloaded from the back end versus checklists that have been completed locally on the client but are not yet synced. The parameter is necessary to make logic decisions on the client as checklists that have been completed and synced to the back end are no longer allowed to be edited. Do not change this setting unless you are integrating SAP Asset Manager with another product besides ASPM.

The *CHECKLISTS* parameters correspond to the rules found in the OData mobile data object *SAM1911_ASPM_CHECKLIST*. You can add a data filter rule to your customer namespace, or change the existing parameter-rule association to a new parameter-rule association.

Procedure

1. Using the ConfigPanel, navigate to *Mobile Application Configuration* > *Parameters tab*. In the left column, *Defined Mobile Applications*, select your application.

   The *Parameter List* populates with a list of all parameters available for the application.
2. You can scroll down to find the *CHECKLISTS* group parameters, listed in the *Context* section, or perform a search using the *Search* box. Highlight the parameter you want to configure and click the *Change* button.



3. Make your desired parameter association changes, or change the value of a parameter to *Z*, a custom activity catalog type.
4. Check the `<Active>` flag to ensure that the parameter is used by the mobile application. If desired, and if not already checked, check the `<No Runtime Change>` box to ensure that the value of the parameter is not overridden at runtime through synchronization processing.
5. *Save* your changes.

Results

You have enabled the checklist parameters in the ConfigPanel.

Next Steps

Continue to the following procedures to finish configuring the checklist feature for ASPM:

- Setting up an RFC Destination for ASPM Checklist Integration [page 121]
- Mapping the Asset Central Key to a Back End System Key [page 123]

7.2 Setting up an RFC Destination for ASPM Checklist Integration

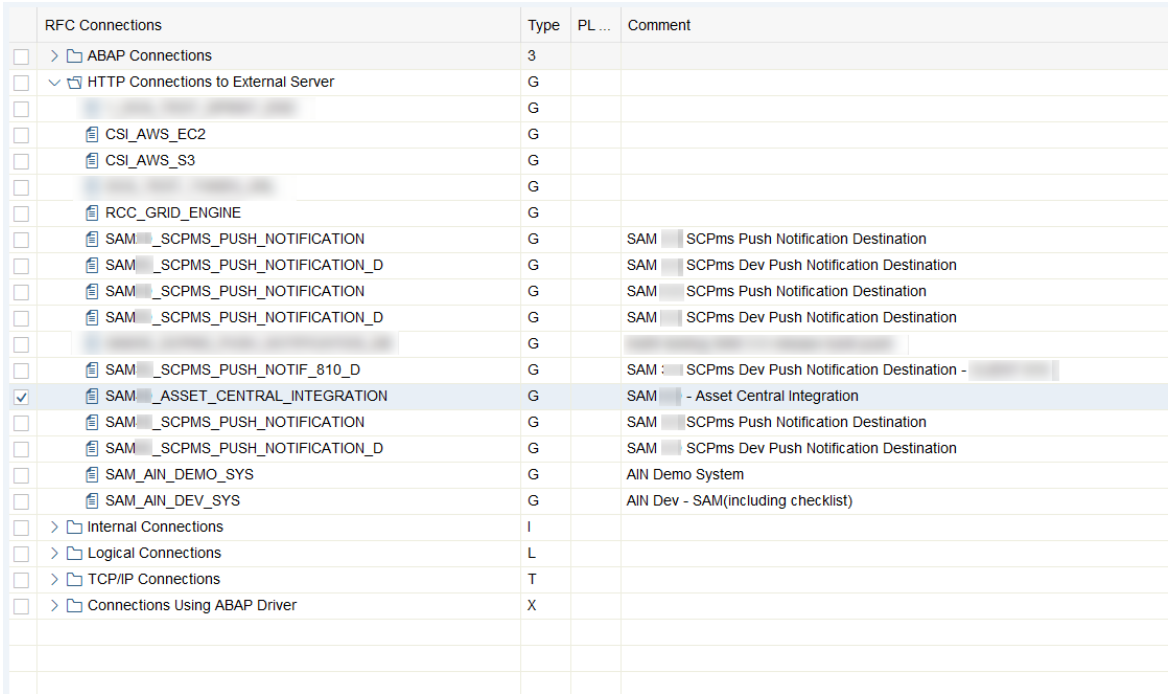
While the RFC destination name filter is already created for you out of box in the ConfigPanel for SAP Asset Manager, it is your responsibility to connect it to the back-end system.

Context

Use the following procedure to connect the existing RFC destination name, *SAM1911_ASSET_CENTRAL_INTEGRATION* to your back-end system.

Procedure

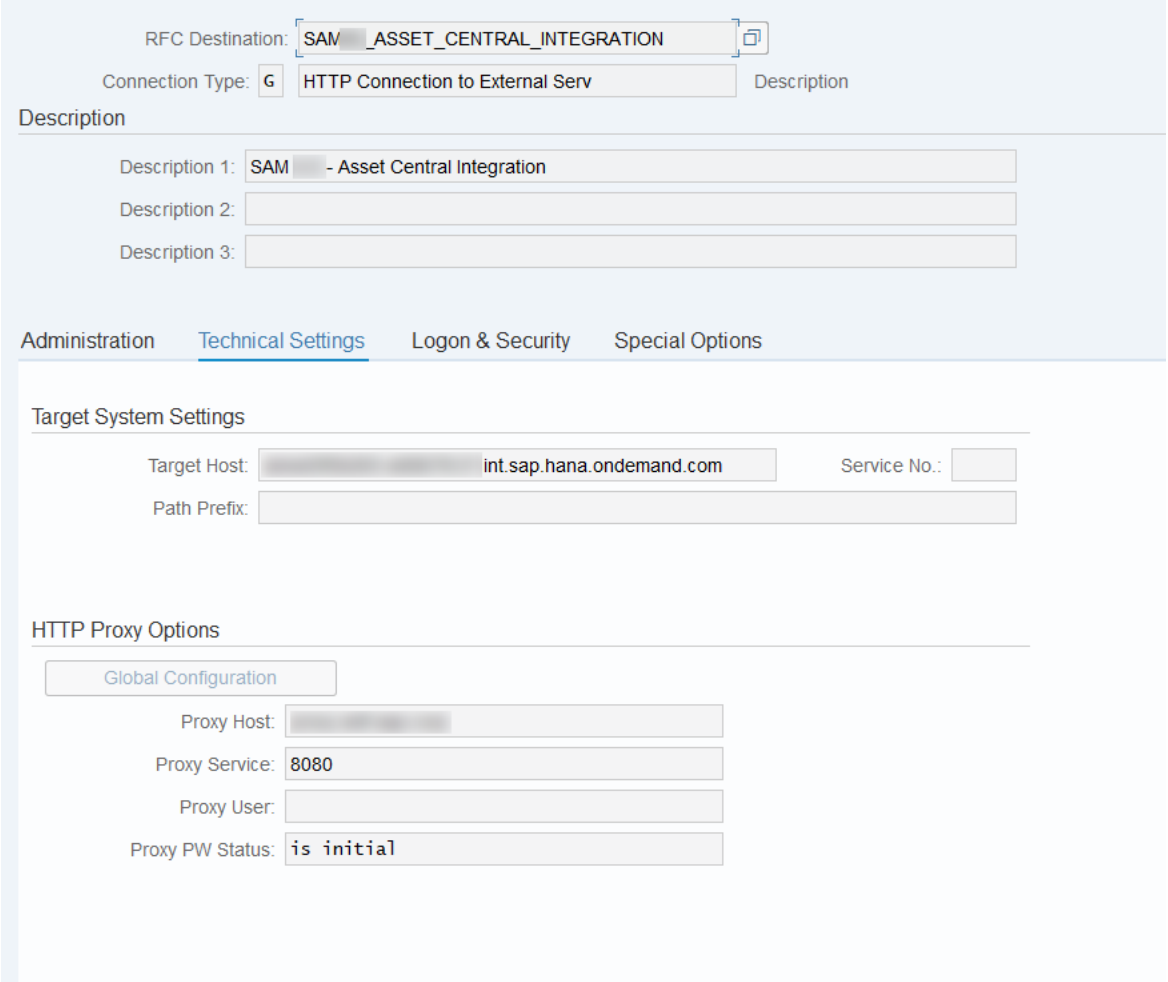
1. In the SAP GUI, using transaction *SM59*, add the following new RFC destination: **SAM1911_ASSET_CENTRAL_INTEGRATION** of type *G (HTTP Connection to External Serv)*



RFC Connections	Type	PL ...	Comment
ABAP Connections	3		
HTTP Connections to External Server	G		
[redacted]	G		
CSI_AWS_EC2	G		
CSI_AWS_S3	G		
[redacted]	G		
RCC_GRID_ENGINE	G		
SAM_SCPMS_PUSH_NOTIFICATION	G		SAM SCPms Push Notification Destination
SAM_SCPMS_PUSH_NOTIFICATION_D	G		SAM SCPms Dev Push Notification Destination
SAM_SCPMS_PUSH_NOTIFICATION	G		SAM SCPms Push Notification Destination
SAM_SCPMS_PUSH_NOTIFICATION_D	G		SAM SCPms Dev Push Notification Destination
[redacted]	G		[redacted]
SAM_SCPMS_PUSH_NOTIF_810_D	G		SAM : SCPms Dev Push Notification Destination - [redacted]
SAM_ASSET_CENTRAL_INTEGRATION	G		SAM - Asset Central Integration
SAM_SCPMS_PUSH_NOTIFICATION	G		SAM SCPms Push Notification Destination
SAM_SCPMS_PUSH_NOTIFICATION_D	G		SAM SCPms Dev Push Notification Destination
SAM_AIN_DEMO_SYS	G		AIN Demo System
SAM_AIN_DEV_SYS	G		AIN Dev - SAM(including checklist)
Internal Connections	I		
Logical Connections	L		
TCP/IP Connections	T		
Connections Using ABAP Driver	X		

2. On the *Technical Settings* tab of the new connection, set the *Target Host* to match the ASPM API of the SAP Cloud Platform mobile services. The `<Service Number>` and `<Path Prefix>` are not required for checklist integration.

> **i Note**
>
> If necessary, configure the proxy that you are using to allow your back-end systems to connect to the Internet.



3. On the *Logon & Security* tab, under the *Logon Procedure*, select *Basic Authentication*. Enter the user name and password of the service user.
4. In the *Security Options* section of the *Logon & Security* tab, ensure that the *SSL* is set to *Active*.
5. Select your SSL certificate from the certificates existing on the `<SSL Certificate>` list.
6. Save the connection and perform a connection test. If the configuration is completed properly, a 404 HTTP response is returned.

Next Steps

Continue to Mapping the Asset Central Key to a Back End System Key [page 123]

7.3 Mapping the Asset Central Key to a Back End System Key

Ensure you map the `/MERP/AIN_OBJ_LINK_SERV_BADI` BAdI from your Asset Central system to your back end system.

Context

If you are installing SAP Asset Manager with the ASPM checklist feature as a 4.0 or above system, you do not have to perform the following configuration, as it is configured out of box. If you are upgrading from an earlier release, you will need to map the Asset Central key to a SAP S/4HANA back end key or a SAP ERP back end key.

Procedure

1. Using the SAP GUI, run transaction *SE18*.
2. Implement the BAdi class `/MERP/AIN_OBJ_LINK_SERV_BADI` and add your own mapping.
3. *Save* your changes.

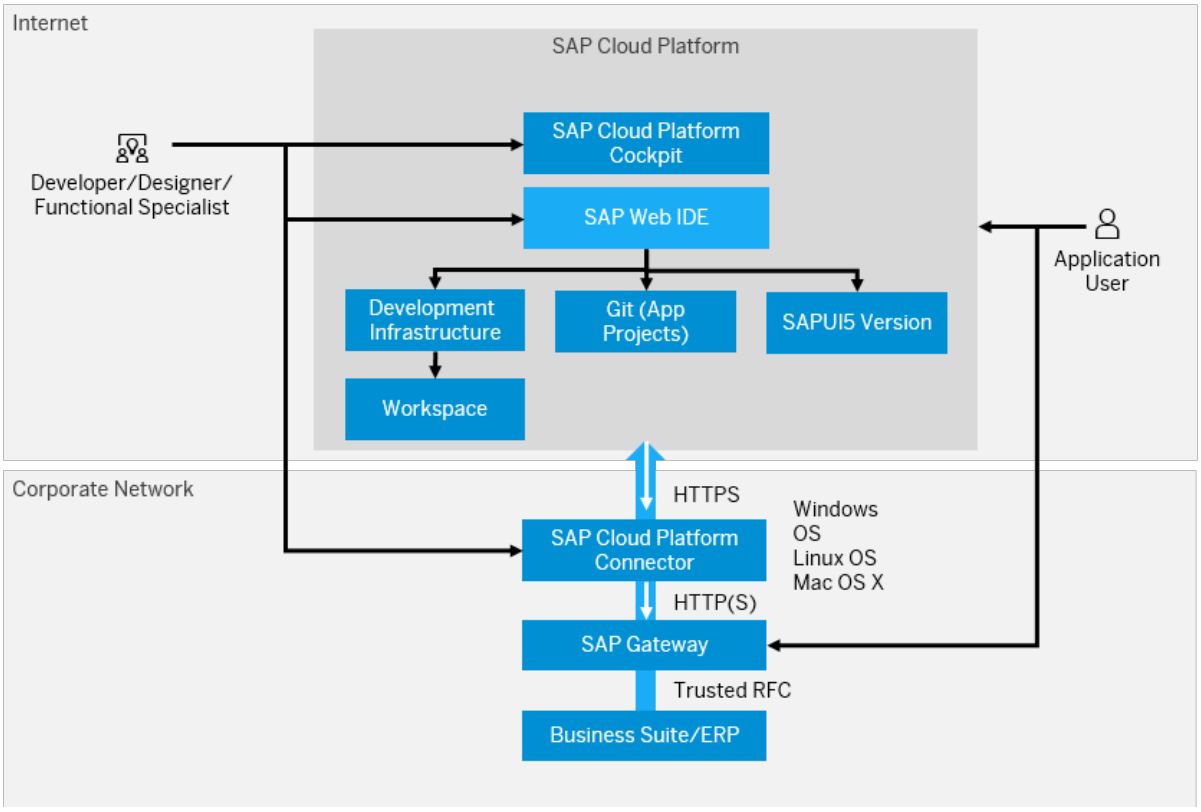
8 SAP Web IDE

SAP Web IDE is a browser-based IDE consisting of integrated parts that interact with each other and with an SAP system.

SAP Web IDE Full-Stack streamlines the end-to-end application lifecycle – easily develop, test, build, deploy, and extend role-based, consumer-grade apps for business users. Create applications rapidly and deliver an outstanding user experience. Developers can extend or build SAP Fiori apps, create SaaS solutions, extend SAP S/4HANA cloud services, develop hybridmobile applications, and build IoT apps for SAP Leonardo, using the UI development toolkit for HTML5 (SAPUI5) for desktop and mobile devices, SAP HANA toolset, and Java programming language and technologies. Since SAP Web IDE Full-Stack runs on , it needs no installation and allows you to integrate with other services that run on the platform – such as SAP Fiori Cloud apps, Git integration, mobile services, IoT services, and more.

Architecture

The following diagram provides high-level typical architecture for SAP Web IDE Full-Stack.



Component	Description
	enables customers and partners to rapidly build, deploy, and manage cloud-based enterprise applications that complement and extend your SAP or non-SAP solutions, either on-premise or on-demand.

Component	Description
cockpit	Central point for managing all activities associated with your account and for accessing key information about your applications. For more information, see SAP Cloud Platform cockpit.
SAP Web IDE application	Integrated development environment used to create or extend SAP UI5 or SAP Fiori applications.
Git	Revision control and source code management system.
SAPUI5	User interface technology that is used to build and adapt client applications. For more information, see UI development toolkit for HTML5 - Demo Kit
connector	Allows SAP Web IDE and to connect to an on-premise system securely and with minimal configuration effort. For more information, see SAP Cloud Platform connector.
SAP Gateway	Provides a simple way to connect SAP Web IDE to an external SAP system with access to OData functionality.

> **i Note**
>
> When working in SAP Web IDE, the following operations may be processed by our partner Infrastructure-as-a-Service (IaaS) providers:
>
> - Code completion
> - Code validation
>
> These operations may involve transfer and process of data in different regions.
>
> The list of operations is subject to change without prior notice.

Who is it for?

SAP Web IDE is a flexible tool for developers who want to dive right into the code editor without having to spend time configuring and administering the development environment.

The tool is aimed at developers who need a modern and secure environment to create new or extend existing SAP Fiori, SAPUI5, or hybrid applications. Developers are provided with a comprehensive set of tools, including strong code editors with templates, wizards, beautifier capabilities, code completion, code snippets, code validation, code checking, WYSIWYG, and many more features.

> **i Note**
>
> SAP Web IDE does not support touch capabilities.

8.1 Customizing Apps Using the Mobile Development Kit

The Mobile Development Kit for SAP Cloud Platform mobile services is a metadata-based application development platform.

The Mobile Development Kit (MDK) lets you customize, deploy, and manage your customized iOS and Android apps in the cloud. The Mobile Development Kit editor lets you edit various aspects of your application using the Mobile Development Kit editor. It also provides native client support and consumes mobile services such as onboarding, offline OData, life-cycle management, and supportability through the using the Mobile Development Kit client.

The Mobile Development Kit allows business process experts to customize the app in a cloud-based editor using the SAP Web IDE, and developers to edit code directly in the metadata files.

The end-to-end use case for Mobile Development Kit includes tasks involving the following roles:

- Administrator
- Business process expert
- Developer
- User

8.1.1 Customizing Metadata using the Mobile Development Kit

One of the main purposes of the Mobile Development Kit is to easily customize and redeploy metadata for your SAP Asset Manager application.

A typical metadata customization procedure is as follows. This example assumes that metadata definitions already exist in the Mobile Development Kit and that you are customizing them, or changing them:

1. Locate the object you want to modify. You can modify pages, actions, or rules. See the following topics and subtopics for more information on how to create and modify the following metadata objects using the Mobile Development Kit:
    - Create Pages
    - Create Actions
    - Create Rules
2. Deploy the metadata. See Deploying App Metadata from Editor to Mobile Services for more information.

9 Data Distribution Model Overview

A data distribution model defines how and what back end data are downloaded to the mobile devices.

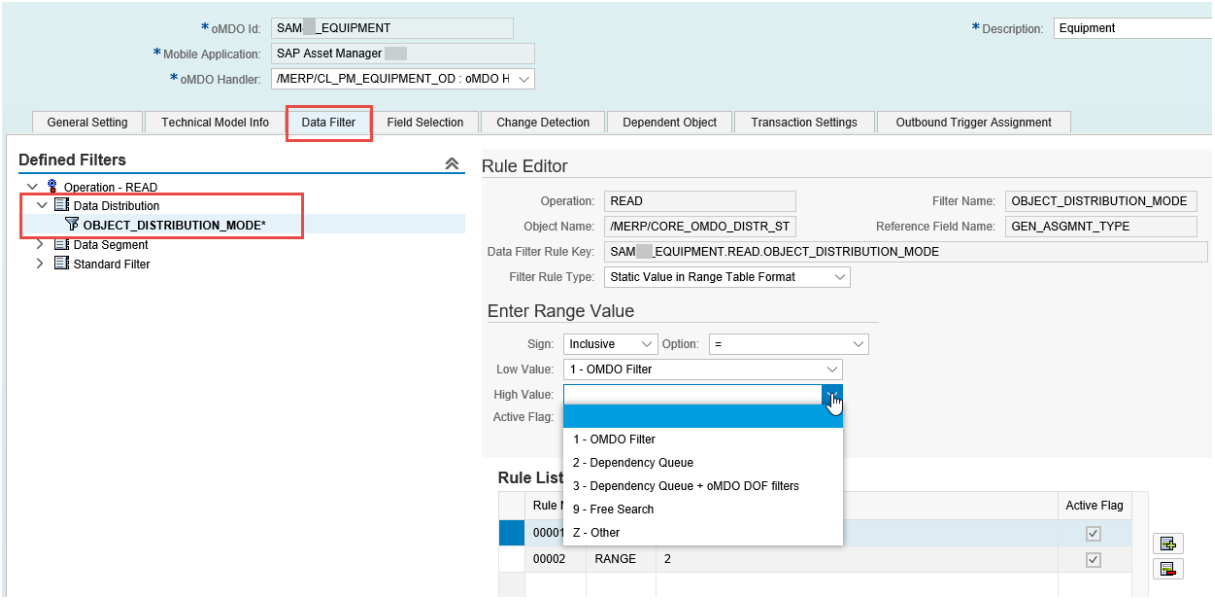
Data distribution models consider various factors when determining what backend data should be downloaded to the mobile client and to the mobile user. Some common criteria are:

- Relevant business processes and business rules
- User business roles in the organization
- Last mobile client synchronization time with the back-end system
- Backend data changes and the time of the changes
- Current state of data objects on the mobile device

For the initial synchronization from the mobile device to the back-end system, the first two bullet points are considered when determining what data should be downloaded to the mobile device and for the requesting user. For subsequent delta synchronizations from the mobile device to the back-end system, all bullet points are considered when determining what data should be downloaded to the mobile device for the requesting user.

The following data distribution models are supported for the SAP Asset Manager application:

- **OMDO Filters**
  Object data collection entirely depends on *OMDO* filter conditions.
- **Dependency Queue**
  Object data collection entirely depends on Dependency Queue objects, and no filter conditions are applied for the fetch criteria.
- **Dependency Queue + OMDO DOF Filters**
  Object data collection is based on dependency queue objects, and the *OMDO DOF* filters are applied for the result set.
- **Other (Custom BAdI)**
  You can implement your own distribution logic using a BAdI.



By default, the SAP Asset Manager application determines object data collection based on distribution model by using *OMDO* filters. In some cases, both *OMDO* filters and *Dependency Queue* are used depending on the business use case. You can choose the appropriate distribution model based on your specific business processes and requirements.

To change the data distribution model for a particular *OMDO* object, complete the steps below:

1. On the ConfigPanel home page, choose *OData Mobile Data Object Configuration*.
   Make sure you select your desired mobile application in the *Mobile Application Filter* field at the top of the page.
2. From the *OData Mobile Data Object List* select the desired OMDO object, such as *SAM1911_EQUIPMENT*, and then click on the *Data Filter* tab.
3. Expand the *Defined Filters* list under *Operation - READ* > *Data Distribution* > *OBJECT_DISTRIBUTION_MODE*. Choose the *Change* button from the menu.
4. Set the distribution model.
5. Save your changes.

9.1 Data Distribution by Assignment Types

By default, the SAP Asset Manager application determines the assignment of work orders and notifications using the personnel number assignment at header level. However, implementation environments in different industries or business types may use a different assignment model from the default to determine the proper technician assignment for work orders and notifications. The SAP Asset Manager application supports several assignment models; you only need to change the assignment type configuration for the specific model.

See Business Object Distribution by Assignment Model [page 75] for more details about assignment model distribution, and how to change assignment type for both work order and notification.

9.2 Common Filters for SAP Asset Manager

The filters listed in the following table are common to all SAP Asset Manager distribution rules. See the specific rules for details on filter requirements for those rules.

Common Filters: Back End Data Filter Matrix

Common Filters: Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution, Mandatory	See specific rule for value	Defines which distribution model is used
COMP_CODE	Data Distribution, Optional	Any	Restricts work order distribution based on the maintenance plant company code on the work order.
CO_AREA	Data Distribution, Optional	Any	Restricts work order distribution based on work order maintenance plant controlling area.
DATE_CLOSE	Standard Filter, Optional	Any	Restricts work order distribution based on date range within which work order has been closed.
DATE_COMPLETION	Standard Filter, Optional	Any	Restricts work order distribution based on date range within which work order has been technically completed.
DATE_RELEASE	Standard Filter, Optional	Any	Restricts work order distribution based on date range within which work order has been released
ORDER_CATG	Data Distribution, Optional	See specific rule for value	Restricts work order distribution based on work order category. For maintenance orders, it should be value `30`.
ORDER_TYPE	Data Distribution, Optional	Any	Restricts work order distribution based on work order type.
ORDERID	Data Distribution, Optional	Any	Restricts work order distribution to a specific work order number.
OPER_ACTTYPE	Standard Filter, Optional	Any	Restricts work order distribution based on work order operation level activity type.
OPER_CONTROL_KEY	Standard Filter, Optional	Any	Restricts work order distribution based on the operation level control key on the work order.

Filter Name	Type	Value	Comments
OPER_EXCL_SYST_STAT	Standard Filter, Optional	Any	Restricts work order distribution by excluding work orders with operations that have a specified system status code.
OPER_EXCL_USER_STAT	Standard Filter, Optional	Any	Restricts work order distribution by excluding work orders with operations that have a specified user status code.
OPER_INCL_SYST_STAT	Standard Filter, Optional	Any	Restricts work order distribution by requiring work order operations with a specified system status code.
OPER_INCL_USER_STAT	Standard Filter, Optional	Any	Restricts work order distribution by requiring work order operations with a specified user status code.
OPER_PLANT	Standard Filter, Optional	Any	Restricts work order distribution based on the operation level plant of the work order.
PLANNER_GROUP	Data Distribution, Optional	Any	Restricts work order distribution based on the header planner group of the work order.
PLANNING_PLANT	Data Distribution, Optional	Any	Restricts work order distribution based on the work order header planning plant.
PLANT	Data Distribution, Optional	Any	Restricts work order distribution based on the header maintenance plant of the work order.
PM_PHASE	Data Distribution, Optional	Any	Restricts work order distribution based on the maintenance process phase of the work order.
WO_EXCL_SYST_STAT	Standard Filter, Optional	Any	Restricts work order distribution by excluding work orders header with a specified system status code.
WO_INCL_SYST_STAT	Standard Filter, Optional	Any	Restricts work order distribution by requiring a work order header with a specified system status code.
WO_EXCL_USER_STAT	Standard Filter, Optional	Any	Restricts work order distribution by excluding a work order header with a specified user status code.
WO_INCL_USER_STAT	Standard Filter, Optional	Any	Restricts work order distribution by requiring a work order header with a specified user status code.
MAIN_WORK_CENTER	Data Distribution, Optional	Any	Restricts work order distribution based on the work order header work center.

Filter Name	Type	Value	Comments
DOC_GOS_RELTYPE	Standard Filter, Optional	Data Segment, Optional	Determines whether the GOS attachment is supported based on a GOS relationship.
DMS_DOC_TYPE	Standard Filter, Optional	Data Segment, Optional	Determines whether the DMS attachment is supported based on the DMS document type.
DOC_LINK_OBJ	Standard Filter, Optional	Data Segment, Optional	Determines whether the DMS attachment is supported based on the linked SAP object.

9.3 Work Order Distribution Rules

The standard SAP Asset Manager application work order distribution is controlled by the OMDO (OData mobile data object) **SAM1911_WORK_ORDER_GENERIC** *READ* operation. It supports several data distribution models for the work order.

You can choose the appropriate distribution model based on your specific business processes and requirements.

9.3.1 Distribution by Work Order Header Person Responsible

Requirements

The following are requirements before configuring the distribution model for Distribution by Work Order Header Person Responsible:

- Mobile user has an employee number (personnel number) assigned in SAP
- Employee number is assigned to the work order header as the person responsible
- Work order has been released
- Work order has not been marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	1	Defines which distribution model is used

Filter Name	Type	Value	Comments
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, it should be value 30.

9.3.2 Distribution by Work Order Operation Person Responsible

Requirements

The following are requirements before configuring the distribution model for Distribution by Work Order Header Person Responsible:

- Mobile user (i.e., the technician) must have an employee number (personnel number) assigned in SAP
- Employee number is assigned to the work order operation as the person responsible
- Work order is released
- Work order is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	2	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.3.3 Distribution by Work Order Suboperation Person Responsible

Requirements

The following are requirements before configuring the distribution model for Distribution by Work Order Suboperation Person Responsible:

- Mobile user (i.e., the technician) must have an employee number (personnel number) assigned in SAP
- Employee number is assigned to the work order suboperation as the person responsible
- Work order is released
- Work order is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	3	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.3.4 Distribution by Capacity Requirement Person Responsible

Requirements

The following are requirements before configuring the distribution model for Distribution by Capacity Requirement Person Responsible:

- Mobile user (i.e., the technician) must have an employee number (personnel number) assigned in SAP
- Employee number is assigned to the work order capacity requirement split records as the person responsible
- Work order is released
- Work order is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	4	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.3.5 Distribution by Work Order Header Planner Group

Requirements

The following are requirements before configuring the distribution model for Distribution by Work Order Header Planner Group:

- Mobile user (i.e., the technician) has been assigned to the planner group based on the business
- Employee number is not required
- Planner group associated with the mobile user is assigned to the work order header
- Work order is released
- Work order is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	5	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.3.6 Distribution by Work Order Operation Work Center

Requirements

The following are requirements before configuring the distribution model for Distribution by Work Order Operation Work Center:

- Mobile user (i.e., the technician) has been associated with a work center in business
- Employee number is not required
- Work center associated with the mobile user is assigned to work order operation
- Work order has been released
- Work order has not been marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	6	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.3.7 Distribution by Work Order Header Business Partner

Requirements

The following are requirements before configuring the distribution model for Distribution by Work Order Header Business Partner:

- Predefined partner function is used in work order for work assignment
- Employee number is not required
- By default, the SAP user ID for the mobile user is used for data distribution by partner function
- Work order is released
- Work order is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	7	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.3.8 Distribution by Work Order Header Work Center

Requirements

The following are requirements before configuring the distribution model for Distribution by Work Order Header Work Center:

- Mobile user (i.e., technician) has been associated with a work center based on the business
- Employee number is not required
- Work center associated with the mobile user is assigned to the work order header
- Work order is released
- Work order is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	8	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.3.9 Distribution through MRS Scheduling Engine

Requirements

The following are requirements before configuring the distribution model for Distribution through MRS Scheduling Engine:

- MRS has been implemented in the SAP system, and is responsible to schedule and update work order capacity records with the assigned technician
- Employee number is required for the mobile user
- Work order is released
- Work order is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	A	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.3.10 Distribution by Free Search

Requirements

The following are requirements before configuring the distribution model for Distribution by Free Search:

- Free search criteria for the work order. Used for an OnDemand work order look-up scenario.
- Employee number is not required
- Work order is released
- Work order is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
WO_ASSIGNMENT_TYPE	Data Distribution Mandatory	7	Defines which distribution model is used
ORDER_CATG	Data Distribution Optional	30	Restricts work order distribution based on work order category. For maintenance orders, value should be *30*.

9.4 Notification Distribution Rules

The OMDO (OData mobile data object) **SAM1911_NOTIF_ASSIGNMENT_TYPE** *READ* operation controls the standard SAP Asset Manager application notification distribution. It supports several data distribution models for the notification.

You can choose the appropriate distribution model based on your specific business processes and requirements.

Notification requests are assigned to the technician directly or assigned through the work center, planner group, or related business partner of the technician. The SAP Asset Manager application supports these different assignment types while downloading notifications associated with the technician.

Valid assignment types are:

- **1 - Header Level Person Responsible:** Assign this notification to the HR personnel number of the technician through the notification header *Partner* section.
- **2 - Notification Task Level Personnel Number:** Assign this notification to the HR personnel number of the technician through individual *Task Personnel Number* field.
- **3 - Header Level Planner Group:** Assign this notification to the planner group associated with the technician through the header level *Planner Group* field.
- **4 - Header Level Business Partner:** Assign this notification to the business partner associated with the technician through header level *Partner Function Maintenance*. The business partner can be anyone related to the notification partner function and associated with the technician, such as user responsible, sold-to-party, or other party. If there is no MAM configuration set up for the user, the default configuration uses *VU-User Responsible* as the default partner function and the technician *SAP User ID* as the partner number.
- **5 - Header Level Work Center:** Assign this notification to the work center associated with the technician through the header level *Work Center* field.

Customers can choose the appropriate distribution model based on their specific business processes and requirements.

9.4.1 Distribution by Notification Header Person Responsible

Requirements

The following are requirements before configuring the distribution model for Distribution by Notification header Person Responsible:

- Technician has an employee number (personnel number) assigned in SAP
- Employee number is assigned to the notification header partner overview as the person responsible
- Notification is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
NOTIF_ASSIGNMENT_TYPE	Data Distribution Mandatory	1	Defines which distribution model is used

9.4.2 Distribution by Notification Task Level Person Responsible

Requirements

The following are requirements before configuring the distribution model for Distribution by Notification task level Personal Responsible:

- Technician has an employee number (personnel number) assigned in SAP
- Employee number is assigned to the notification task *Person Responsible* field
- Notification is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
NOTIF_ASSIGNMENT_TYPE	Data Distribution Mandatory	2	Defines which distribution model is used

9.4.3 Distribution by Notification Header Level Planner Group

Requirements

The following are requirements before configuring the distribution model for Distribution by Notification header level Planner Group:

- Technician is assigned to the planner group based on the business
- Employee number is not required
- Planner group associated with the mobile user is assigned to the notification header
- Notification is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
NOTIF_ASSIGNMENT_TYPE	Data Distribution Mandatory	3	Defines which distribution model is used

9.4.4 Distribution by Notification Header Level Business Partner

Requirements

The following are requirements before configuring the distribution model for Distribution by Notification header level Business Partner:

- Business partner is assigned to the notification header partner overview
- The SAP user ID of the mobile user is used for data distribution by partner function using *User Responsible*
- Employee number is not required
- Notification is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
NOTIF_ASSIGNMENT_TYPE	Data Distribution Mandatory	4	Defines which distribution model is used

9.4.5 Distribution by Notification Header Level Work Center

Requirements

The following are requirements before configuring the distribution model for Distribution by Notification header level Work Center:

- Technician is associated with a work center in business
- Employee number is not required
- Work center associated with the mobile user is assigned to the notification header
- Notification is not marked for deletion

Back End Data Filter Matrix

Filter Name	Type	Value	Comments
NOTIF_ASSIGNMENT_TYPE	Data Distribution Mandatory	5	Defines which distribution model is used

9.4.6 Distribution by Free Search - Notifications

Requirements

The following are requirements before configuring the distribution model for Distribution by Free Search:

- Free search for notification used for an on-demand notification look-up scenario
- Employee number is not required
- Notification is not marked for deletion

Back End Data Filter Matrix

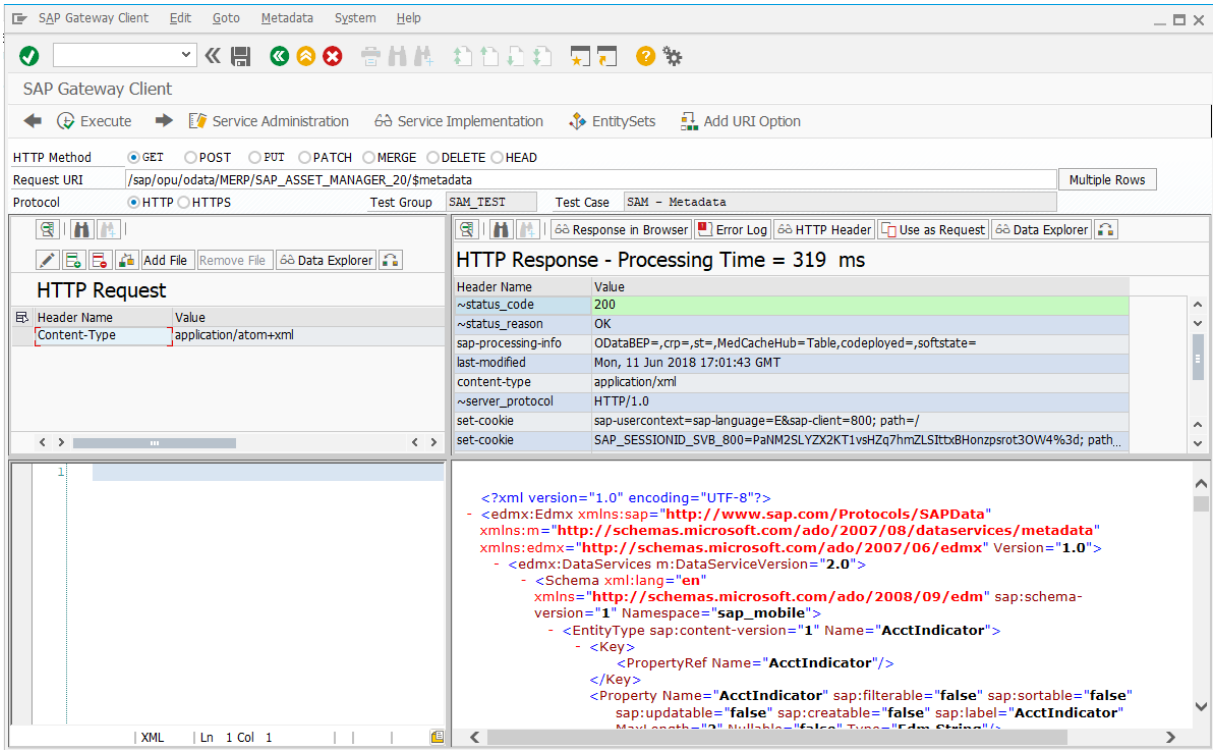
Filter Name	Type	Value	Comments
NOTIF_ASSIGNMENT_TYPE	Data Distribution Mandatory	9	Defines which distribution model is used

10 Troubleshooting

This section describes the various troubleshooting activities that you can perform in error situations, or the app users can perform on a regular basis to ensure the smooth running of the mobile application. It is also explains how to monitor the different components of SAP Gateway, how to use the logs, and how to carry out maintenance activities.

10.1 SAP Gateway Client

You can use the SAP Gateway Client (transaction code: */IWFND/GW_CLIENT*) to test your OData service provider without an OData consumer, such as the SAP Asset Manager mobile client. This tool is especially useful to test your OData service from the back end to identify service-related issues before a service is used by the mobile application.



For more information about how to work with the SAP Gateway Client, see SAP Gateway Client in the SAP Gateway Technical Operations Guide.

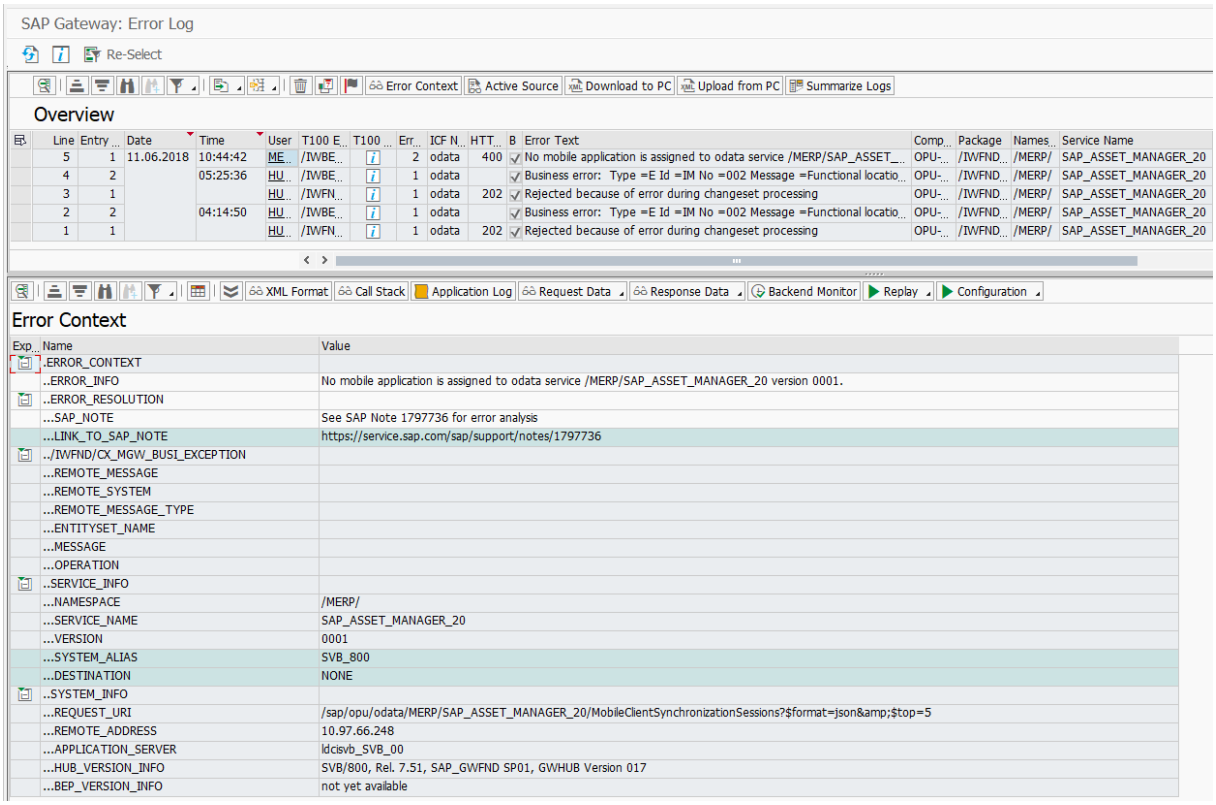
10.2 SAP Gateway Error Logs

Error logs provide detailed context information about errors that have occurred at runtime, enabling you to perform root cause analysis, as well as reproducing and correcting errors.

You can launch the error log with transaction */IWFND/ERROR_LOG* in Gateway Hub systems. Launch the error log with transaction */IWBEP/ERROR_LOG* in your back-end system.

The SAP Gateway error logs reveal basic details about errors and show errors from all users for a given client. Business logic errors are often displayed in this error log due to improper business logic. Other errors displayed include the HTTP code to indicate the type of error.

Note that based on the security level setting, advanced details or the replay function may be hidden or disabled. Note also that these error logs will not show generic authorization errors if users fail to properly authenticate.



You can navigate to different sections from the *Error Context* area as shown above. Choose *Replay* to reproduce and correct errors. Choose from the following two replay options:

- SAP Gateway Client
- Web Browser

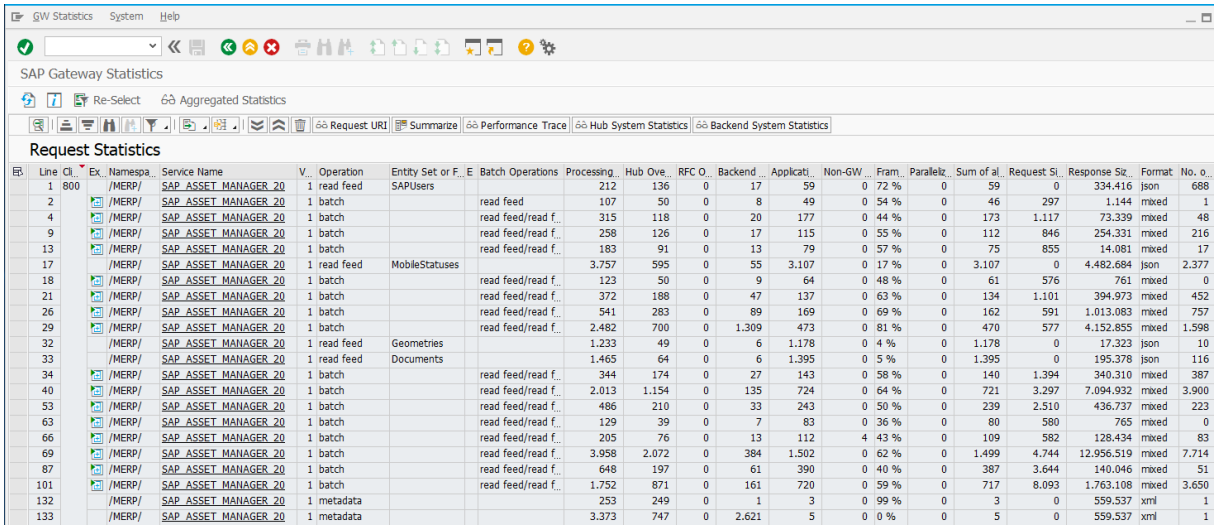
Use option *SAP Gateway Client* to reproduce runtime situations that led to a particular error without accessing the application from the actual mobile client, and to simulate a service at runtime to identify and resolve potential issues.

For more information about how to configure the error log, see Configuration Settings for the Error Log in the *SAP Gateway Technical Operations Guide*.

In addition, use the *Application Log Viewer* to display more technical error details by using transaction */IWFND/APPS_LOG*.

10.3 SAP Gateway Statistics

You can use the SAP Gateway Statistics (transaction code: */IWFND/STATS*) to display the request statistics and aggregated statistics. Each successful OData request has an entry in the statistics records, which is kept for 7 days by default, however, you can extend the period to 30 days. Request statistics can be aggregated, in which case they are kept for 90 days by default, however, you can extend the period to 365 days.



Line	Cli..	Ex..	Namespa..	Service Name	V..	Operation	Entity Set or F..	E	Batch Operations	Processing..	Hub Ove..	RFC O..	Backend ..	Applicati..	Non-GW ..	Fram..	Paralleliz..	Sum of al..	Request Si..	Response Siz..	Format	No. o..
1	800		/MERP/	SAP_ASSET_MANAGER_20	1	read feed	SAPUsers			212	136	0	17	59	0	72 %	0	59	0	334.416	json	688
2			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed	107	50	0	8	49	0	54 %	0	46	297	1.144	mixed	1
4			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	315	118	0	20	177	0	44 %	0	173	1.117	73.339	mixed	48
9			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	258	126	0	17	115	0	55 %	0	112	846	254.331	mixed	216
13			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	183	91	0	13	79	0	57 %	0	75	855	14.081	mixed	17
17			/MERP/	SAP_ASSET_MANAGER_20	1	read feed	MobileStatuses			3.757	595	0	55	3.107	0	17 %	0	3.107	0	4.482.684	json	2.377
18			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	123	50	0	9	64	0	48 %	0	61	576	761	mixed	0
21			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	372	188	0	47	137	0	63 %	0	134	1.101	394.973	mixed	452
26			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	541	283	0	89	169	0	69 %	0	162	591	1.013.083	mixed	757
29			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	2.482	700	0	1.309	473	0	81 %	0	470	577	4.152.855	mixed	1.598
32			/MERP/	SAP_ASSET_MANAGER_20	1	read feed	Geometries			1.233	49	0	6	1.178	0	4 %	0	1.178	0	17.323	json	10
33			/MERP/	SAP_ASSET_MANAGER_20	1	read feed	Documents			1.465	64	0	6	1.395	0	5 %	0	1.395	0	195.378	json	116
34			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	344	174	0	27	143	0	58 %	0	140	1.394	340.310	mixed	387
40			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	2.013	1.154	0	135	724	0	64 %	0	721	3.297	7.094.932	mixed	3.900
53			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	486	210	0	33	243	0	50 %	0	239	2.510	436.737	mixed	223
63			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	129	39	0	7	83	0	36 %	0	80	580	765	mixed	0
66			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	205	76	0	13	112	4	43 %	0	109	582	128.434	mixed	83
69			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	3.958	2.072	0	384	1.502	0	62 %	0	1.499	4.744	12.956.519	mixed	7.714
87			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	648	197	0	61	390	0	40 %	0	387	3.644	140.046	mixed	51
101			/MERP/	SAP_ASSET_MANAGER_20	1	batch			read feed/read f..	1.752	871	0	161	720	0	59 %	0	717	8.093	1.763.108	mixed	3.650
132			/MERP/	SAP_ASSET_MANAGER_20	1	metadata				253	249	0	1	3	0	99 %	0	3	0	559.537	xml	1
133			/MERP/	SAP_ASSET_MANAGER_20	1	metadata				3.373	747	0	2.621	5	0	0 %	0	5	0	559.537	xml	1

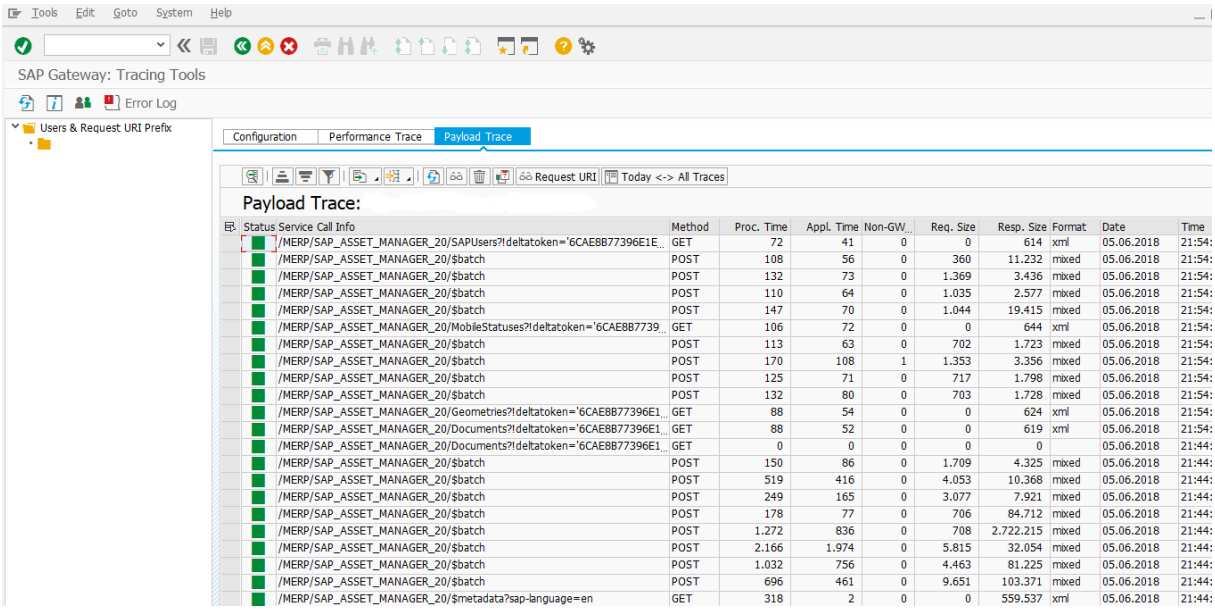
SAP Gateway Statistics aggregates the entries by various entities, for example, client, namespace, service name & version. With the */IWFND/STATS* transaction you can verify details, such as processing time, response size by entity, and other statistics about the complete request.

10.4 SAP Gateway Tracing Tools

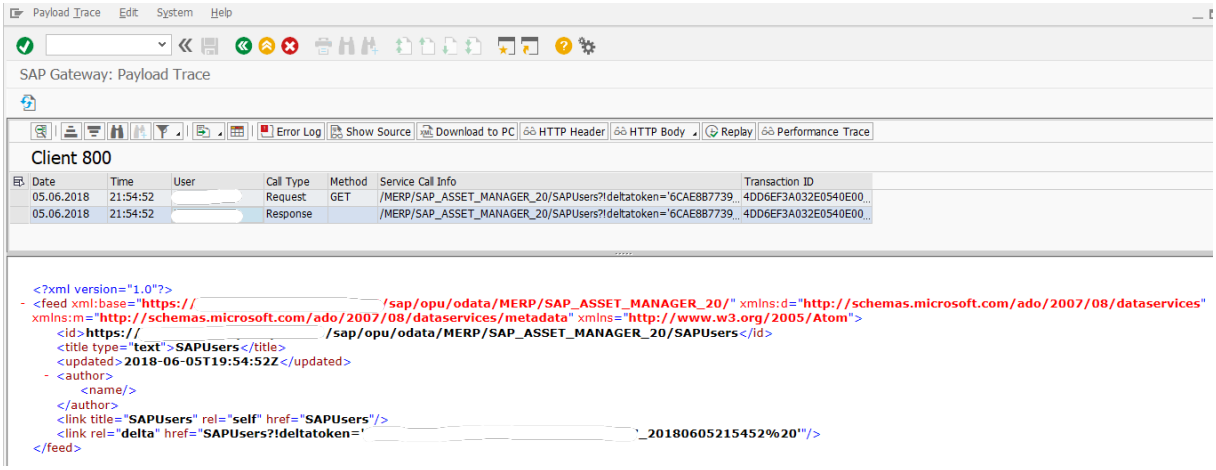
The SAP Gateway provides tracing tools (transaction code: `/IWFND/TRACES`) to trace on a particular user for both performance and payload.

Performance trace enables you to monitor performance at service call level for both the SAP Business Suite and the SAP Gateway. Payload trace enables you to monitor the service calls with request and response data, and to replay and simulate the service calls without accessing the application from the mobile client.

Traces display detailed request and response data coming into the SAP Gateway. Traces are active for only a short time, and are purged on a regular basis.



With this tool, you can verify the exact content of the request header and body that is sent from the mobile device, and also check the response from the SAP Gateway.



For information about how to configure and activate the payload trace tool, see Tracing Tools: Configuration in the *SAP Gateway Technical Operations Guide*.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

www.sap.com/contactsap

THE BEST RUN SAP