



**SAP SuccessFactors** 

**PUBLIC**

Document Version: 1H 2025 – 2025-04-11

# Implementing the Metadata Framework (MDF)

# Content

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Overview</b> . . . . .  | <b>7</b>  |
| 1.1      | What Is the Metadata Framework? . . . . .                                    | 7         |
| 1.2      | Features. . . . .  | 7         |
| 1.3      | Prerequisites. . . . .   | 8         |
| 1.4      | Optimistic Locking. . . . .  | 9         |
| 1.5      | Important Configuration Guidelines and Usage Limits. . . . .                 | 10        |
| <b>2</b> | <b>Implementation Sequence</b> . . . . .                                     | <b>14</b> |
| <b>3</b> | <b>Enabling MDF</b> . . . . .  | <b>15</b> |
| 3.1      | Enabling Through Provisioning. . . . .                                       | 15        |
| 3.2      | Enabling Through Upgrade Center. . . . .                                     | 16        |
| 3.3      | Configuring Permissions for an MDF User Role. . . . .                        | 17        |
| <b>4</b> | <b>Generic Object Definition</b> . . . . .                                   | <b>19</b> |
| 4.1      | Creating a New Object Definition. . . . .                                    | 19        |
|          | Basic Configurations in Object Definition. . . . .                           | 20        |
| 4.2      | Configuring the Object Definition. . . . .                                   | 25        |
|          | Adding Fields. . . . .   | 25        |
|          | Adding Associations. . . . .   | 33        |
|          | Adding Workflows. . . . .  | 35        |
|          | Adding Searchable Fields. . . . .  | 36        |
|          | Adding Business Key Fields. . . . .  | 37        |
|          | Adding Security. . . . .   | 38        |
|          | Adding Rules. . . . .  | 39        |
|          | Adding Translation Labels for Country/Region and Currency. . . . .           | 43        |
|          | Creating MDF Blocks. . . . .   | 43        |
| 4.3      | Understanding Object Definitions and Relationships. . . . .                  | 45        |
|          | Field Criteria. . . . .  | 45        |
|          | Associations in MDF. . . . .   | 47        |
|          | List of MDF Core Objects. . . . .  | 52        |
|          | Auto-Number ExternalCode for MDF Objects. . . . .                            | 53        |
| 4.4      | Adding Object Definitions to a Bundle for Transportation. . . . .            | 55        |
| 4.5      | Additional Object Configurations. . . . .                                    | 56        |
| 4.6      | Legislatively Sensitive Personal Data Configuration. . . . .                 | 60        |
| 4.7      | Configuring Fields for Read Access Logging. . . . .                          | 61        |
| 4.8      | Custom Field Limits on Pre-delivered and Custom MDF Generic Objects. . . . . | 62        |

|          |   |           |
|----------|---|-----------|
| <b>5</b> | <b>Managing MDF Object Instance.</b>                                  | <b>64</b> |
| 5.1      | Creating an MDF Object Instance.                                      | 64        |
| 5.2      | Viewing MDF Object Data.  | 65        |
| 5.3      | Copying Configuration and Non-User Transactional MDF Data.            | 67        |
| 5.4      | Configuring the Disable Searchable Fields Config in Auto Complete.    | 68        |
| 5.5      | Referring to an RBP Subject User Field on the Referred Node.          | 70        |
| 5.6      | Forward Propagation.  | 74        |
| 5.7      | Improved Attachment Handling for MDF.                                 | 77        |
| 5.8      | Viewing MDF Object Data Volumes.                                      | 78        |
| 5.9      | Adding Object Instances to a Transport Bundle.                        | 80        |
| <b>6</b> | <b>Manage Configuration UI.</b>                                       | <b>82</b> |
| 6.1      | Creating a Configurable UI.   | 83        |
| 6.2      | Adding Configurable UI to a Bundle for Transportation.                | 84        |
| 6.3      | Linking User Interface to Object Definition.                          | 85        |
| 6.4      | Known Limitations and Considerations.                                 | 85        |
| <b>7</b> | <b>MDF Configurations for Latest People Profile.</b>                  | <b>87</b> |
| <b>8</b> | <b>Setting Up Security for Object Instances.</b>                      | <b>89</b> |
| 8.1      | Setting Up Security for MDF Using Role-Based Permissions (RBP).       | 89        |
|          | Overview.   | 89        |
|          | Reporting.  | 90        |
|          | Enabling Security for Objects.  | 90        |
|          | Enabling Security for Fields.   | 91        |
|          | Enabling Security for Parent Object Associations.                     | 92        |
|          | Enabling Security for Composite Child Permissions.                    | 93        |
|          | Order of Permissions.   | 94        |
| 8.2      | Restricting Access.   | 94        |
|          | Assigning Roles to Groups.  | 94        |
|          | Restricting Data Access of a Role with Target Population or Criteria. | 95        |
|          | Enabling Search Field Restriction.                                    | 97        |
| <b>9</b> | <b>MDF Picklists.</b>   | <b>99</b> |
| 9.1      | Searching for a Picklist.   | 100       |
| 9.2      | Adding Picklists to a Bundle for Transportation.                      | 100       |
| 9.3      | Creating a Picklist.  | 102       |
| 9.4      | Editing a Picklist.   | 105       |
| 9.5      | Picklist Values.  | 106       |
|          | Creating and Editing a Picklist Value.                                | 106       |
|          | Sort Order for Picklist Values.                                       | 108       |
|          | Deactivating Picklist Values.   | 108       |
| 9.6      | Picklist Versions.  | 110       |

|           |  |            |
|-----------|--|------------|
|           | Creating a Picklist Version. . . . .   | 110        |
|           | Deactivating a Picklist Version. . . . .                                       | 111        |
| 9.7       | Deactivating a Picklist. . . . .   | 112        |
| 9.8       | Cascading Picklists. . . . .   | 113        |
| 9.9       | Deleting Picklists. . . . .  | 116        |
|           | Deleting Picklist or Picklist Versions. . . . .                                | 117        |
|           | Deleting Picklist Values. . . . .  | 118        |
| 9.10      | Importing and Exporting Picklists. . . . .                                     | 119        |
|           | Exporting an MDF Picklist. . . . .   | 119        |
|           | MDF Picklist Import. . . . .   | 119        |
| 9.11      | Making Picklist Value Fields Visible and Editable in Picklist Center. . . . .  | 122        |
| 9.12      | Disabling Sync of Non-Unique External Code. . . . .                            | 123        |
| 9.13      | Field Mapping from Legacy to MDF Picklists. . . . .                            | 124        |
| <b>10</b> | <b>Importing and Exporting Data. . . . .</b>                                   | <b>127</b> |
| 10.1      | Exporting MDF Object Data. . . . .   | 127        |
| 10.2      | Configuring RBP to Allow User to View Exports in Monitor Jobs. . . . .         | 128        |
| 10.3      | Exporting UI Configurations. . . . .   | 129        |
| 10.4      | Importing MDF Data. . . . .  | 131        |
|           | Importing Data Purging Options: Incremental or Full Purge. . . . .             | 132        |
|           | Configuring Batch Size for Importing MDF Entities. . . . .                     | 137        |
|           | Suppress Duplicate Inserts. . . . .  | 137        |
|           | Using Delimit Functionality for MDF Entities. . . . .                          | 139        |
|           | Valid Values for Enum and Boolean Data Type Fields in Import Template. . . . . | 145        |
|           | Validating MDF Data. . . . .   | 146        |
|           | Deleting Records for MDF Objects Through Import. . . . .                       | 147        |
|           | Adding Attachments to MDF Objects During Data Import. . . . .                  | 149        |
| 10.5      | Importing Associated Objects. . . . .  | 150        |
| 10.6      | Rule Integration with MDF Import. . . . .                                      | 151        |
|           | Initialize Rules. . . . .  | 151        |
|           | Validate Rules. . . . .  | 152        |
|           | Save Rules. . . . .  | 152        |
| <b>11</b> | <b>Integrating with Business Rules. . . . .</b>                                | <b>153</b> |
| 11.1      | Business Rules in MDF. . . . .   | 153        |
|           | Configuring a Business Rule for MDF. . . . .                                   | 154        |
|           | Assigning a Rule to an MDF Object. . . . .                                     | 158        |
|           | Order of MDF-Based Rule Execution. . . . .                                     | 159        |
| 11.2      | Editing the Rule Type Picklist. . . . .  | 160        |
| 11.3      | Rules for Composite Child Entities. . . . .                                    | 161        |
| 11.4      | Assigning Rule Execution Criteria for an MDF Object. . . . .                   | 162        |
| 11.5      | Importing, Exporting, and Transporting Rules. . . . .                          | 163        |

|           |   |            |
|-----------|---|------------|
|           | Exporting Rules. . . . .  | 163        |
|           | Importing Rule-Related Generic Objects. . . . .                     | 164        |
|           | Importing Rules. . . . .  | 165        |
|           | Constraints on Rules Imports. . . . .                               | 166        |
|           | Adding a Rule to a Transport Bundle. . . . .                        | 167        |
| 11.6      | MDF Rule Scenario Use Cases. . . . .                                | 168        |
|           | Providing Default Values. . . . .                                   | 169        |
|           | Validating User Input Text or Values. . . . .                       | 170        |
|           | Setting Field Values Based on User Input. . . . .                   | 172        |
|           | Generating Alert Notifications. . . . .                             | 173        |
|           | Configuring Rule-Based Workflow. . . . .                            | 175        |
|           | Populating Calculable Fields on the User Interface. . . . .         | 177        |
|           | Updating Field Values Based on Changes on Another Field. . . . .    | 179        |
|           | Preventing Deletion of Specific Records. . . . .                    | 180        |
|           | Changing Field Attributes on the User Interface. . . . .            | 181        |
|           | Excluding Rule Execution during Import. . . . .                     | 183        |
|           | Validate Rules Based on Next, Previous, or Original Values. . . . . | 184        |
| <b>12</b> | <b>Workflows in MDF. . . . .</b>                                    | <b>186</b> |
| 12.1      | Creating a New Workflow. . . . .                                    | 187        |
| 12.2      | Need to Know About Workflows in MDF. . . . .                        | 188        |
| 12.3      | Supported Workflow Actions. . . . .                                 | 189        |
| 12.4      | MDF Workflows on the Home Page. . . . .                             | 190        |
| <b>13</b> | <b>MDF Alerts. . . . .</b>  | <b>194</b> |
| 13.1      | Configure MDF Alerts. . . . .                                       | 194        |
|           | Creating an Object Definition. . . . .                              | 195        |
|           | Creating an Alert Message. . . . .                                  | 195        |
|           | Creating an Instance for Multiple Alerts. . . . .                   | 196        |
|           | Creating a Basic Workflow. . . . .                                  | 197        |
|           | Configuring a Business Rule. . . . .                                | 198        |
|           | Configuring Post Save Rules for an Object. . . . .                  | 199        |
|           | Creating Records for the Object. . . . .                            | 200        |
| <b>14</b> | <b>Viewing Admin Alerts for MDF Object Data Volumes. . . . .</b>    | <b>202</b> |
| <b>15</b> | <b>Reporting. . . . .</b>   | <b>204</b> |
| 15.1      | Overview. . . . .   | 204        |
| 15.2      | Prerequisites. . . . .  | 204        |
| 15.3      | Viewing MDF Objects from Analytics. . . . .                         | 205        |
| <b>16</b> | <b>Purging MDF Data (Non-DRTM). . . . .</b>                         | <b>206</b> |
| <b>17</b> | <b>MDF OData API. . . . .</b>                                       | <b>209</b> |

|           |   |            |
|-----------|---|------------|
| 17.1      | Exposing Custom MDF Objects to OData API. . . . .                         | 209        |
|           | API Subversioning. . . . .  | 211        |
| 17.2      | MDF Foundation Object Entities. . . . .                                   | 212        |
| 17.3      | Data Mapping Between MDF and OData API. . . . .                           | 213        |
| 17.4      | Locale Handling in MDF Entities. . . . .                                  | 224        |
| 17.5      | MDF OData API Operations. . . . .   | 229        |
|           | Permissions. . . . .  | 230        |
|           | Query Operation. . . . .  | 231        |
|           | Edit Operations. . . . .  | 254        |
| <b>18</b> | <b>Migrating MDF Configurations Using the Instance Sync Tool. . . . .</b> | <b>276</b> |
| 18.1      | Instance Sync Tool. . . . .   | 276        |
|           | Transferring Configurations Between Instances. . . . .                    | 276        |
|           | Transferring Generic Objects. . . . .                                     | 277        |
|           | Transferring Configurable UIs. . . . .                                    | 277        |
|           | Transferring Object Definitions. . . . .                                  | 277        |
|           | Transferring MDF Rules. . . . .   | 278        |
| 18.2      | Test Sync/Run Sync. . . . .   | 279        |
| 18.3      | Monitoring a Sync Operation. . . . .                                      | 280        |
| 18.4      | Integrating with Instance Sync Tool. . . . .                              | 281        |
| 18.5      | Integrating with SuccessStore. . . . .                                    | 283        |
| 18.6      | Troubleshooting the Instance Sync Tool. . . . .                           | 284        |
| <b>19</b> | <b>Change History. . . . .</b>  | <b>285</b> |

# 1 Overview

## 1.1 What Is the Metadata Framework?

The SAP SuccessFactors Metadata Framework (MDF) is an in-app extensibility platform, used to build new and simple applications with ease, without writing any code.

MDF allows customizing and extending predelivered MDF-based applications such as Position Management, Time Off, JDM 2.0, Picklist Center and so on, that suit your requirements. It empowers customers to create and maintain custom extensions called MDF custom objects or Generic Objects (GO) within SAP SuccessFactors. MDF is integrated with features such as Role-based Permissions (RBP), Business Rules, OData and People Profile, and allows the use of their capabilities seamlessly.

MDF is universally available for all SAP SuccessFactors customers to build such simple extensions and make use of all features as long as customer usage is within the specified limits as mentioned in [Important Configuration Guidelines and Usage Limits \[page 10\]](#).

MDF allows you to capture basic details for which data doesn't grow substantially over a period of time. For example, an application to track the Covid-19 Vaccination Status of each employee. However, MDF extensions aren't meant for full-scale application development such as building complete clock-in/clock-out applications or putting all historical payroll data into custom objects. For such use cases, SAP recommends either using the SAP-delivered solution or making use of the SAP Business Technology Platform

### Related Information

[SAP Business Technology Platform](#)

## 1.2 Features

MDF provides a set of features that enable you to build end-to-end applications. Using MDF, you can also add objects to the MDF data model or modify the behavior of existing objects to suit your client's needs.

Using MDF, you can do the following:

- Configure, define, and manage objects in the data model using an easy-to-use UI
- Add custom fields, not available in the standard instance, to your client's instance of existing objects to extend an object's scope of access
- Add custom parent-child associations catering to your client's instance to out-of-the-box MDF objects
- Add instance-specific business logic to an object using rules
- Add instance-specific workflow logic to out-of-the-box MDF objects



- Create custom rules specific to your instance using an easy-to-use UI
- Create custom workflows specific to your instance using an easy-to-use UI
- Create instances of newly or previously defined MDF objects
- Create a configurable UI for your instance
- Add security to your client's MDF objects and data using RBP
- Expose MDF objects and data through Common Object OData REST APIs to add programming logic
- Create custom UIs using MDF internal SCAs or Common Object OData REST APIs
- Create ODS reports or ad hoc reports for your client
- Import and export MDF data and object definitions

## 1.3 Prerequisites

This section deals with the settings or permissions that you need to set before you can define MDF objects in your client's instance.

### Administrative Access

Since using MDF involves modification of the SAP SuccessFactors metadata, you must have administrative access to the instance for which MDF is being enabled. You must also have a good understanding of data model and business rules in order to understand the consequences of changes to the metadata.

#### 📌 Note

You don't need any additional SKUs or licenses to use MDF if you have access to SAP SuccessFactors HCM suite. Although MDF can be used without Employee Central, some features in MDF require Employee Central license.

### Naming Convention

Before you create new entries in your database using MDF, you must establish a consistent and easy-to-understand naming convention for your objects.

- In most cases, the names you create can be up to 30 characters long; in some cases they can be longer.
- In all names, use alphanumeric characters, spaces, and symbols, but you must not use a space as the first character.
- You find that a descriptive approach to naming locations, departments, and other database entities helps users to understand and maintain the system more easily. For example, naming an internal site *Boston Customer Support* is more meaningful than naming it *Site1*.



## Enable Language Packs

In *Provisioning*, in *Edit Company Settings*, select ► *Company Settings* ► *Language Packs* ► *English US (English US)* ►. Enabling this setting avoids application errors such as Configure Business Rule, Manage Data, Manage Business Configuration, Manage Configuration UI and so on.

### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

## 1.4 Optimistic Locking

In a multiuser scenario, when two or more users try to modify the same data, control of data concurrency and data consistency plays a vital role. To avoid inconsistencies which result in issues during integration, Optimistic Locking Support for MDF Delivered Objects has been implemented.

Optimistic locking prevents data from being corrupted or invalidated when multiple users try to update the same object in the database. Optimistic Locking has been implemented for **BusinessUnit**, **Division**, **Department**, **Position** and Generic Objects used in Business Configuration UI.

To ensure better control of data consistency and data concurrency in case of bulk imports, if there is an optimistic lock exception on one of the records, the whole batch is rolled-back.

### ⓘ Note

There is a configuration option provided by MDF in Admin Center where depending on custom defined number of records per batch (default being 500 and minimum being 20), the file gets divided into several batches.

By reducing the number of records in such a batch, the importing time per batch will decrease and also minimizes the probability of parallel object changes.

## 1.5 Important Configuration Guidelines and Usage Limits

Before you start implementing MDF, keep the following configuration guidelines and usage limits in mind. These guidelines and limits are set forth by the Metadata Framework to help provide you guardrails, avoid potential errors, and improve system performance.

### MDF Configuration Guidelines and Usage Limits

| Area  | Guidelines and Limits  | More Information  |
|---|--|---|
| Custom MDF objects                                  | 250 (Default)<br>400 (Maximum)   | To increase the limit, raise a request with Technical Support and provide a business reason specifying why you need a higher limit.<br><br><a href="#">Creating a New Object Definition [page 19]</a>   |
| Custom fields for pre-delivered MDF generic objects | 30 string fields<br>30 number fields<br>20 decimal fields<br>10 date fields  | For more information about adding custom fields, refer to <a href="#">Adding Fields [page 25]</a> .<br><br><b>Note</b><br>Object-specific limits may be available. Please refer to the product documentation of that object for more details.   |
| Fields for custom generic objects                   | Based on the internal configuration, custom MDF objects will have either of these sets of limits:<br><br>75 string fields<br>65 number fields<br>60 decimal fields<br>30 date fields<br><br>Or<br><br>150 string fields<br>100 number fields<br>100 decimal fields<br>75 date fields | <b>Note</b><br>You get a warning message when 80% of the limit is reached and an error message when the limit is exceeded.<br><br>Although objects might be allowed to have higher limits, it is always recommended to keep the limits even below the lower limits to avoid performance issues.<br><br>For more information about adding custom fields, see <a href="#">Adding Fields [page 25]</a> |
| Data volume   | Up to 50 million records for all custom MDF objects together   | Large data volume can impact system performance or even lead to application errors.<br><br>For more information, see <a href="#">Creating an MDF Object Instance [page 64]</a>  |

| Area                         | Guidelines and Limits  | More Information  |
|------------------------------|--|---|
| Number of time slices        | 1000   | <p>Applicable for both predelivered and custom MDF objects.</p> <p>Adding time slices increases the time needed to load data. In applications where pagination isn't available, the number of time slices can impact system performance.</p> <p>Large number of time slices also means increased time in saving data during import and OData integration.</p> <p>If you're using picklists outside MDF, keep only one single time slice with a past effective date as early as possible, for example, 1/1/1900.</p>   |
| Number of picklist versions  | 10 for a single picklist   |   |
| Number of composite children | Records per Composite Child <ul style="list-style-type: none"> <li>• Recommended: 500</li> <li>• Hard limit: 3000</li> </ul>   | <p>Applicable for both predelivered and custom MDF objects. It's not applicable to OData import and export, which supports a much larger number.</p> <p>We recommend that you observe the limits to ensure system performance and avoid potential data display issues.</p> <div data-bbox="833 919 1427 1110" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <p>At any given time, the total number of records of all composite children associated to a single parent object must not exceed 3000 records.</p> </div> <p>For more information, see <a href="#">Associations in MDF [page 47]</a>.</p> |
| Workflows                    | <ul style="list-style-type: none"> <li>• Avoid using the <i>Workflow Routing</i> field to configure workflows. Use business rules instead.</li> <li>• Avoid adding multiple rules to trigger workflows. Use IF, THEN, and ELSE statements in a single rule for different conditions.</li> <li>• Set <i>Pending Data</i> to "Yes" for objects with workflow and "No" for objects without workflow. Exceptions are allowed only if instructed in documentation.</li> </ul> | <a href="#">Workflows in MDF [page 186]</a>   |

| Area  | Guidelines and Limits   | More Information   |
|---|---|--|
| Effective dating and MDF version history          | Understand the difference between MDF version history and effective dating and choose the proper feature for your extension scenario.   | <p>MDF version history is used to capture audit information that can be pulled into a change audit report.</p> <p>Effective dating allows you to create multiple records for a given external code and multiple records for a given date (effective date).</p> <p>For more information, see <a href="#">Basic Configurations in Object Definition [page 20]</a>.</p>   |
| Data purge  | <p>Purge MDF data regularly, including:</p> <ul style="list-style-type: none"> <li>• Audit data</li> <li>• Business data</li> <li>• Attachments</li> </ul>  | <p>By purging MDF data that you no longer need, you can reduce cloud storage and improve system performance. For more information about data purge, see <a href="#">Purging MDF Data (Non-DRTM) [page 206]</a>.</p>  |
| Changes to MDF core objects                       | Don't change object definitions of MDF core objects except the labels for fields, associations, and object definition.  | <p>Changing the definition of MDF core objects can lead application errors.</p> <p>For more information, see <a href="#">List of MDF Core Objects [page 52]</a>.</p>   |
| Concurrent edits                                  | Avoid concurrent edit operations on the same object.  | <p>Concurrent edit can lead to data errors.</p> <p>If you're performing multiple edit operations through import, Integration Center, or other means, please ensure that:</p> <ul style="list-style-type: none"> <li>• They aren't performed concurrently on the same object type.</li> <li>• If they're performed in parallel, the data pertaining to the same external code or business key must be within the same operation/job. Editing the same record in different operations concurrently can cause data errors.</li> </ul> |
| File size for data import                         | <ul style="list-style-type: none"> <li>• When using CSV or ZIP files in <a href="#">Import and Export Data</a>, the file size limit is 500 MB.</li> <li>• When using FTP, the file size limit is 1 GB.</li> </ul> | <p>For files larger than the limits, split the file into smaller pieces and import in batches.</p>   |
| Number of picklist values for a version/timeslice | 50,000  | <p>We recommend that you keep the number of picklist values for a given version or timeslice under 50,000. More picklist values can cause performance issues during add, update, or delete operations. If you need more values, please evaluate whether a custom MDF object can be used instead of a picklist.</p> <p>For more information, see <a href="#">MDF Picklists [page 99]</a>.</p>   |

| Area  | Guidelines and Limits   | More Information  |
|---|---|---|
| Configuration UI                                    | Business rules and RBP permissions are recommended.   | <ul style="list-style-type: none"> <li>• Business Rules are recommended over the UI rules.</li> <li>• It is recommended to control field visibility using Role-based Permission (RBP), instead of using of the UI rules.</li> <li>• However, there are certain use cases where only UI rules can be used <ul style="list-style-type: none"> <li>• Hide or Show field(s) or UI group(s) based on a field value change.</li> <li>• Making a field Mandatory or Non-Mandatory.</li> </ul> </li> <li>• Rules to hide/show fields in a grid layout must be avoided, as it results in distorting the UI.</li> <li>• Rules to make fields Mandatory / Non-Mandatory in a grid layout causes UI to become slower</li> </ul> |
| Configuration UI: UI Rules                          | Maintain the overall number of UI rules below 25  | <ul style="list-style-type: none"> <li>• It is recommended to keep less number of UI rules.</li> <li>• Adding more conditions to reduce the number of UI rules does not help.</li> <li>• If multiple fields need to be hidden/shown based on the same condition, group them together using UI group and hide /show the complete group rather than changing the individual fields.</li> <li>• Every addition of UI rules adds increased complexity resulting in performance downgrade and bad user experience.</li> </ul>  |
| Configuration UI: Layout                            | <p>The People Profile page does not support Grid Layout.</p> <p>It is recommended to use Flow or Form Layout</p>                                      |   |
| Number of Composite Association                     | Recommended limit of composite association is 5   | You can have more composite associations for a given object. However, with increased number of child records the UI and save/update operations becomes slower. In extreme cases it might stop working.  |
| Level of Composite association                      | The limit of composite association level is 2 (parent and 2 levels) for custom MDF objects and 3 (parent and 3 levels) for pre-delivered MDF objects. | We will support the available level of composites for existing pre-delivered and custom MDF objects and also allow creation of new child that doesn't exceed the existing level of composite for a given root object.   |
| Number of Records for <b>Valid When</b> association | The recommended limit for <b>Valid When</b> association record is 100   | With the increased number of records on UI, Import, or APIs, the <b>Save</b> operation will become slower. In extreme cases, it might stop working.   |

## 2 Implementation Sequence

Following this sequence of steps will take you from activating MDF in your instance to creating a baseline generic object definition.

1. Enable MDF in Provisioning.

### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

2. Set permissions to use the MDF.
3. Configure an object definition.
  1. Create a new object definition to launch the object definition setup.
  2. Configure the object definition to add basic object information to the definition.
  3. Complete the object definition form.
  4. Add workflow routing to add routing information.
  5. Add fields to add custom fields to the object definition.
  6. Add rules to add business logic using rules to the object definition.
  7. Add associations to add parent-child relations with other objects.
  8. Add searchable fields to add keyword searches.
  9. Add security to add object and/or field level security to the object definition.
4. Create the picklists.
5. Create the MDF object instance.

# 3 Enabling MDF

## 3.1 Enabling Through Provisioning

This section explains how to set up the client to use MDF.

### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

### Enabling MDF

1. In Provisioning, in *Edit Company Settings*, select *Company Settings*.
2. Select the *Enable Generic Objects* checkbox.

### Enabling Workflows

While still in Provisioning, you also need to select the following for Workflows:

- Workflow
- Enable the Attachment Manager
- Employee Central Foundation Objects
- Effective Dated Data Platform
- Employee Central V2
- Enable Business Rules for Workflow Derivation

### ⓘ Note

MDF doesn't require additional product licences if the customer has the licence to any one of the SAP SuccessFactors products. Some of the MDF features are dependent on other SAP SuccessFactors products and won't be supported unless the customer has the appropriate product licence.

For example, the workflow isn't supported if the *Employee Central Foundation Objects* feature isn't enabled. If the *Enable the Attachment Manager* feature isn't enabled, the customer cannot use the attachment Data type.

Check on the respective feature licence requirements and ensure if the features are required.

### ⓘ Note

For Workflows to work in your system, the Corporate Data Model also needs to be updated. Following is the sample Workflow Configuration.



```

<hris-element id="wfConfig">
  <label>Workflow Configuration</label>
  <hris-field max-length="32" id="externalCode" visibility="both" required="true">
    <label>Workflow ID</label>
    <label xml:lang="de-DE">Workflowkennung</label>
    <label xml:lang="en-GB">Workflow ID</label>
    <label xml:lang="es-ES">ID de flujo de trabajo</label>
    <label xml:lang="fr-CA">ID du flux de travail</label>
    <label xml:lang="fr-FR">ID du flux opérationnel</label>
    <label xml:lang="it-IT">ID flusso di lavoro</label>
    <label xml:lang="ja-JP">ワークフロー ID</label>
    <label xml:lang="ko-KR">워크플로우 ID</label>
    <label xml:lang="pt-BR">ID do workflow</label>
    <label xml:lang="zh-CN">工作流程 ID</label>
  </hris-field>
  <hris-field max-length="32" id="is-delegate-supported" visibility="both">
    <label>Is Delegate Supported</label>
  </hris-field>
  <hris-field max-length="32" id="escalation" visibility="both">
    <label>Escalation</label>
  </hris-field>
  <hris-field max-length="32" id="name" visibility="both">
    <label>Name</label>
    <label xml:lang="de-DE">Name</label>
    <label xml:lang="en-GB">Name</label>
    <label xml:lang="es-ES">Nombre</label>
    <label xml:lang="fr-CA">Nom</label>
    <label xml:lang="fr-FR">Nom</label>
    <label xml:lang="it-IT">Nome</label>
    <label xml:lang="ja-JP">名前</label>
    <label xml:lang="ko-KR">이름</label>
    <label xml:lang="pt-BR">Nome</label>
    <label xml:lang="zh-CN">名称</label>
  </hris-field>
  <hris-field id="remind-indays" visibility="both">
    <label>Remind In Days</label>
    <label xml:lang="bg-BG">Напомняне в дни</label>
    <label xml:lang="bs-BS">Diingatkan Dalam Hari</label>
    <label xml:lang="bs-ID">Mengingatkan Hari</label>
    <label xml:lang="cs-CZ">Připomenutí za (dny)</label>
    <label xml:lang="da-DK">Påmind om - dage</label>
    <label xml:lang="de-CH">Erinnerung nach (Tage)</label>
  </hris-field>

```

## 3.2 Enabling Through Upgrade Center

The Metadata Framework is used with the modules like SAP SuccessFactors Recruiting Management to build the application. So enabling MDF is prerequisites for such applications.

Some of the new applications which are typically deployed through Upgrade Center uses MDF. If MDF is not yet enabled for these applications, it is possible to enable MDF in the Upgrade Center.

To do this, go to ► [Admin Center](#) ► [Release Center](#) ► [Upgrade Center](#) ► and switch on the *Extension Center* feature available under Optional Upgrades. This enables MDF automatically.

## 3.3 Configuring Permissions for an MDF User Role

You can grant permission to users to create new MDF object definitions in their instance.

### Procedure

1. Go to [Admin Center](#) > [Manage Permission Roles](#).
2. Select a **Permission Role** for which you want to give MDF access.  
**Permission Role Detail** page appears.
3. Click [Permission...](#)
4. In the [Metadata Framework](#) section, specify what permissions users in this role must have.

| Permission                                      | Description   |
|---|---|
| <b>Configure Object Definitions</b>             | Allows you to access Configure Object Definition page to manage objects and related configurations  |
|   | <b>Note</b><br>Enabling this permission also grants additional permission on <b>Pick-list Center</b>  |
| <b>Manage Data</b>                              | Allows you to access Manage Data page   |
| <b>Configure Business Rules</b>                 | Allows you to access Configure Business Rules page  |
| <b>Access to Business Rule Execution Log</b>    | Allows you to configure and download rule execution logs  |
| <b>Access to non-secured objects</b>            | Allows you to access non-secured MDF objects. This permission is required for accessing any non-secured MDF objects in applications where MDF such objects are consumed, such as Manage Data, People Profile, and so on.  |
|   | <b>Note</b><br>You can allow users to access non-secured objects even if they don't have the necessary permissions assigned to them by opting for the <a href="#">Continue to Allow Users Access to Non-Secured Objects without Permissions being Assigned</a> feature in Upgrade Center. |
| <b>Import Permission on Metadata Framework</b>  | Allows you to import and export MDF objects   |
| <b>Admin access to MDF OData API</b>            | Allows you to access MDF OData API  |
| <b>Manage Configuration UI</b>                  | Allows you to access Manage Configuration UI page   |
| <b>Manage Positions</b>                         | Allows you to access Manage Position page   |
| <b>Manage Mass Changes for Metadata Objects</b> | Allows you to access mass change on MDF objects   |
| <b>Manage Sequence</b>                          | Allows you to access Manage Sequence. You can use this field to generate automatic sequence number for a rule or automatic externalCode generation  |

5. In *Manage Foundation Object Types* section, specify all the permissions for **Workflow** foundation object.
6. Assign the ► *Admin Center Permissions* ► *Monitor Scheduled Jobs* ► permission for access to *Scheduled Job Manager*.

#### ⓘ Note

If you want access to results of all MDF jobs, you also need to have the ► *Manage User* ► *Allow users to view all the jobs. (By Disabling this option, users can view only their job status.)* ► permission. Otherwise, you're able to see all MDF jobs but only have access to results of MDF jobs that are submitted by you.

7. Click *Done* to exit.
8. Click *Save Changes* to save your configurations.

MDF is now activated on your instance.

## Related Information

[Assigning Permissions to a Role](#)

[Monitoring Scheduled Jobs in Admin Center](#)

# 4 Generic Object Definition

A generic object definition represents a business entity or object. An object definition contains information about fields (properties of an object), associations (relationship to other objects), workflows, business rules, and security configuration for an object.

Using the Configure Object Definitions interface, you can create new objects or customize your applications by modifying existing object definitions to suit your need.

## Prerequisites

Before you start, ensure you understand object definitions and their relationship with other objects, preferably in an ERD diagram. You also need to know how to map your object data types to the data types supported by MDF.

## 4.1 Creating a New Object Definition

You can create objects to add to your custom object definition using the Configure Object Definition interface.

## Prerequisites

You have read and understand the guidelines and limits in [Important Configuration Guidelines and Usage Limits \[page 10\]](#).

## Context

You can add fields (properties of an object), associations (relationship to other objects), workflows, business rules, and security configuration for the object during creation.

## Procedure

1. Navigate to the Admin Center.
2. In the Tools search field, type *Configure Object Definitions*.
3. In the *Configure Object Definitions* page, from the *Create New* dropdown list, choose *Object Definition*. The Object Definition page is displayed.

4. Enter basic configuration information as described in [Basic Configurations in Object Definition \[page 20\]](#).
5. Choose *Save*.

For more information on how to add fields, associations, workflows, business rules, and security configurations, refer to *Configuring the Object Definition* section.

## Related Information

[Basic Configurations in Object Definition \[page 20\]](#)

[Configuring the Object Definition \[page 25\]](#)

### 4.1.1 Basic Configurations in Object Definition

List of the basic properties you can configure in an object definition.

This table gives an explanation of the fields in the form and why they're important.

| Field | Description   |
|-------|---|
| Code  | A unique code that identifies the object. If the code already exists, you're prompted to enter another.<br><div data-bbox="435 1077 1425 1228"><p><b>Note</b></p><p>We recommend that after creating a custom object, don't modify the code field because it might be used in configurations like rules, configuration UI, role-based permissions, and so on.</p></div> |

| Field            | Description  |
|------------------|--|
| Effective Dating | <p>Specifies whether the object allows effective dating and, if so, of which type. Effective dating allows you to manage and track data on an effective date in the past, at present, or in the future.</p> <p>Available effective dating values are:</p> <ul style="list-style-type: none"> <li>• <i>None</i>: The object is not effective-dated. Choose this option if you want only the current record in the system to be relevant, or if the data doesn't change over time, for example, emergency contacts of an employee, date of birth, etc.</li> <li>• <i>Basic</i>: A record can be changed only once per day. Choose this option if the past, present, and future records of the object are all relevant, for example, the salary information of an employee.</li> <li>• <i>From Parent</i>: Used for composite association only. The effective dating setting of the object is inherited from its parent object. Object with "From Parent" effective dating can't be accessed without the parent object. If you delete a record of the parent object, its associated child records are also deleted.</li> <li>• <i>Multiple Changes Per Day (MCPD)</i>: A record can be changed multiple times per day. Please make sure you have a valid business case before choosing this option.</li> </ul> <p>You can only increase effective dating, that is, from None to Basic or MCPD, or from Basic to MCPD. It must never be decreased, that is, from Basic to None, or from MCPD to Basic or None.</p> <p>You can't change the Effective Dating value of an object definition from None, Basic, or MCPD to From Parent, if the object is associated with another object with a <i>Valid When</i> association.</p> <div data-bbox="431 1020 1419 1325" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <p>Objects with effective dating set to <i>From Parent</i> can't be used as a reference in another object. However, it can be used in composite association. You can't access these objects via a quickcard, Manage Data, or other tools that depend on MDF.</p> <p>Effective dating is not to be confused with audit history. An object can have effective dating set as "none" and still have the changes logged in change audit. However, only the latest change is persisted.</p> </div> |
| API Visibility   | <p>This field is only relevant to custom objects. If you want to expose a custom object to OData V2 as an API, change the field value to <i>Editable</i> or <i>Read Only</i>. Default setting is <i>Not Visible</i>.</p> <div data-bbox="431 1440 1419 1745" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <p>Custom objects created before this field were introduced in November 2014 are set to Editable so that they continue to be visible in the API as before. Custom objects created after November 2014 aren't visible by default and must be manually set.</p> <p>The accessibility of pre-delivered objects via the OData V2 API varies depending on the specific objects. Refer to the API reference guide for detailed information on using pre-delivered objects as APIs.</p> </div>   |

| Field               | Description   |
|---------------------|---|
| Status              | <p>Status of the object. If set to Inactive, the object instance isn't used and not visible anywhere in the system, including the rule engine, configurable UI designer, and role-based permissions.</p> <p>An object can't be deactivated if data exists or if it's referenced by an active association. If an object that is referenced in a rule is deactivated, that rule is invalidated. Data can't be imported or exported for an inactive object.</p>  |
| MDF Version History | <p>The MDF Version History allows you to capture the MDF audit information about added, updated, and deleted field values and records. This type of audit information is captured as change audit and can be viewed in <a href="#">Change Audit Report</a>.</p> <p>This is important in integrations such as Employee Central Payroll and Employee Central to user management.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> <li>• <b>No:</b> If this option is selected, or no selection is made, history records aren't stored for any operation and no audit records are generated.</li> <li>• <b>Delete History:</b> If this option is selected, only the audit records from the delete operation of that object type are generated. The data deleted by a full purge import and translatable fields are also included.</li> <li>• <b>Complete History:</b> If this option is selected, every update or delete operation generates a copy of the current record and stores it as audit information. The data deleted by a full purge import and translatable fields are also included.</li> </ul> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>⚠ Caution</b></p> <p>Do not enable complete MDF version history if you're handling a large data set, such as OData API and MDF import, or if all your records are frequently replicated from an external system. As a best practice, use MDF Audit Data Purge to regularly delete audit data that are no longer needed. For more information, see the Related Information section.</p> </div> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>📌 Note</b></p> <p>You can't use audit information to restore deleted data.</p> <p>Enable MDF version history for the objects you want to capture the audit information. If not enabled, SAP will not be able to provide any audit report through support tickets.</p> <p>By default, <i>MDF Version History</i> for picklists is set to <i>Complete History</i>, so that you can generate a change audit report for all picklist changes.</p> </div> |
| Default Screen      | <p>You can assign a default configurable UI to an MDF object. The assigned default UI is used in all places where the MDF object appears, such as the <a href="#">Manage Data</a> page, the MDF quickcard, and the "Details" popup.</p>   |



| Field              | Description  |
|--------------------|--|
| Label              | <p>Name of the object that appears on the UI.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>You can't modify MDF object and field labels using <a href="#">Admin Center</a> &gt; <a href="#">Text Replacement</a>.</p> </div>  |
| Description        | You can provide description of the object you're creating.   |
| API Sub Version    | <p>This field controls whether the technical fields in the ODATA API are exposed to the customers or not.</p> <p>Set to one of the following:</p> <ul style="list-style-type: none"> <li>No Selection: Defaults to V1.1.</li> <li>V1.0: All system fields are kept in OData.</li> <li>V1.1: All system fields are removed.</li> </ul>  |
| Subject User Field | Specify the fields that have subject user information. This allows you to determine the workflow routing. Only 'User' data type fields are allowed for this configuration. If ExternalCode is defined with data type 'User', then ExternalCode is used as default value.   |
| Workflow Routing   | <p>You can add workflow routing to the object by associating it with a workflow. Workflows are approval processes set up for another user (typically a manager) to approve or reject changes to an employee's data. Workflow routing requires Employee Central Foundation Objects.</p> <p>You can also associate an object with workflow by creating a workflow rule and linking it in the rules section. However, we recommend you to use any one of the two methods.</p> <p>For more information about workflows, see <i>Implementing and Configuring Workflows in Employee Central</i> in <a href="#">SAP Help Portal</a>.</p>  |
| Pending Data       | <p>If you've linked a workflow to an object that involves an approval, and you want the data changes to take effect only if approved, set this field to <a href="#">Yes</a>. If the changes aren't approved, the system rejects the changes. If data updates can be made without any approval via workflow, set this field to <a href="#">No</a>.</p> <p>If the workflow feature is enabled, there's a navigation named wfRequestNav. This allows the workflow request information to be pulled directly into the MDF object.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>MDF workflow Pending History (PH) record is supported in OData.</p> <p>When the HR user imports the data, a workflow goes to the HR Admin (approver) to approve or reject the changes. When the workflow is finally approved, the pending record is saved as a normal record and can be queried in OData.</p> </div> |

| Field           | Description  |
|-----------------|--|
| Todo Category   | <p>If Pending Data is set to <a href="#">Yes</a>, the to-do category value determines where the workflow approval goes for changes made to the object.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <p>This field is read-only. For all custom MDF objects, <a href="#">Generic Objects Change Requests</a> is the default category. For predelivered MDF objects, the value is set by the respective business areas.</p> </div>   |
| Object Category | <p>Object category determines the type of data an object represents. MDF objects are classified into one of the following categories:</p> <ul style="list-style-type: none"> <li>• <b>Uncategorized:</b> When the category of the object isn't determined yet, this option can be used. This is the default category for all custom objects.</li> <li>• <b>Configuration:</b> This category is used when the object represents configuration data. For example, Picklist.</li> <li>• <b>Non-User Transactional:</b> This category is used when the object represents data that isn't related to a user nor a configuration data.</li> <li>• <b>User Transactional:</b> This category is used when the object represents user-related information. For example, Payment Information.</li> <li>• <b>Technical:</b> This category is used when the object represents technical data like integration status, database sync status, and so on.</li> </ul> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• When the Effective Dating of an object is set to From Parent, the parent's object category is automatically inherited by the object.</li> <li>• For an object of type User or RBP Subject User Field, the object category must be set to User Transactional.</li> </ul> </div> |

## Need to Know

- You can add default values for a field in the object definition. Default values of fields of Data Types String, Number, Date, Boolean are assigned during the onInitialize event. The insert operation gets the value from the previous time slice object. Customers can overwrite the default value but they can't remove it.
- An event gets triggered whenever MDF object definitions change. This explicit call helps MDF OData to refresh or update the metadata whenever there's a change in MDF data model, that is, either addition or deletion of object definitions.

If you enable or disable a feature or change the language packs through Provisioning leading to a change that requires OData metadata refresh, the refresh must be done manually.

### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

- MDF supports one-to-one mapping between a parent object and child object. A validation is added when you try to associate one more parent object to a child object, thus not allowing you to complete the action.
- Whenever there's any change to the settings in MDF Object type configuration, a new validation is added. This displays meaningful warning messages that include information on the consequences of changes to MDF object type configuration.
- The Metadata Framework supports multiple alert rules per object. A new column, MDFAlertType is added to the alert\_entry table. To monitor the changes, you have to create MDFAlertType instances and assign the alert rule to it. Ensure that the MDFAlertType is unique for each of these rules.
- You can't delete a parent field if the object definition is part of a composite association.

## 4.2 Configuring the Object Definition

After you create an object definition, you can configure your object definition by adding fields, associations, workflows, rules, and security at object and field levels. You can also configure fields as searchable fields and business key fields.

### 4.2.1 Adding Fields

Add fields to the MDF object definition using the Configure Object Definition page to define the object structure.

#### Prerequisites

You have read and understand the guidelines and limits described in [Important Configuration Guidelines and Usage Limits \[page 10\]](#).

You have created an object definition and you're in the edit mode of the object definition page.

#### Procedure

1. Go to the [Fields](#) section of the object configuration and enter the following information to add a new field:

You can add fields to define the object structure. These fields can have a specific data type, different attributes, and can refer to other MDF objects, foundation objects, or to a User.

##### Note

Choosing which field to add depends on, how you plan to use the object that you're defining. However, [externalCode](#) field is required for each object.

| Field            | Description  |
|------------------|--|
| <b>Name</b>      | Enter a unique name for the field in the current object definition. After creating the object, it can be used in configurations like rules, configuration UI, role-based permissions, and so on. We recommend not to edit any field names after the object is used in any such configurations.   |
| <b>Data Type</b> | <p>Choose the data type from the following supported data types:</p> <ul style="list-style-type: none"> <li>• <b>String:</b> Used for text input fields. MDF allows storing text up to 4000 characters long.</li> <li>• <b>CLOB:</b> Used for text inputs fields when you need to store text more than 4000 characters long.</li> <li>• <b>Number:</b> Used to store integers. Allowed characters are 0–9. Display of the number, such as using a comma separator depends on the user's locale settings.</li> <li>• <b>Auto Number:</b> Used to store numbered lists. The value of this field is autogenerated by the system and it's always a read-only field on the UI.</li> <li>• <b>Decimal:</b> Used to store decimal numbers. Displayed based on the user's locale settings, such as using a comma or decimal point as the decimal separator.</li> <li>• <b>Boolean:</b> The two allowed values are Yes and No.</li> <li>• <b>Date:</b> Used to store a date from a date picker.</li> <li>• <b>Picklist:</b> Used to restrict the field value by providing a pre-defined list of values to select from.</li> <li>• <b>Translatable:</b> Used to allow the entry of localized strings for different locales. You can enter values in this field and maintain translations of the values in different languages. Based on the language selected in your user preferences, the corresponding language text is displayed.</li> <li>• <b>Data Source:</b> Used to allow Java object definitions to specify a class implementing data source interface to provide a set of custom code defined autocomplete values. This data type can't be used for custom object fields. This data type is used only for pre-delivered object fields.</li> <li>• <b>Generic Object</b><br/>Used to reference a generic object. When the object is created, this field shows a value help with a list of the instances of that generic object.</li> <li>• <b>Foundation Object</b><br/>Used to reference a foundation object. When the object is created, this field shows a value help with a list of the instances of that foundation object.</li> <li>• <b>User</b><br/>Used to reference a user object. When the object is created, this field shows a value help with a list of active users. For custom object fields, this doesn't support Employee Central based effective dated user search. However, you can configure it to show an effective dated user list.</li> <li>• <b>DateTime</b><br/>Used to store the date and time information related to a time zone. This is a more granular data type. Time is converted to the time zone of the user who</li> </ul> |

| Field                        | Description   |
|------------------------------|---|
|                              | <p>views or requests the export, and not to that of the user who originally entered the time.</p> <ul style="list-style-type: none"> <li>• Time<br/>Allows the user to enter times without having to specify any time zone information. Times are displayed in HH:mm:ss format and are validated to ensure that a valid time value is entered. However, multiple formats are allowed based on the settings in the object definition, for example, hideSeconds and TwelveHourFormat.</li> <li>• Attachment<br/>Allows the user to upload files and associate them to other MDF, foundation, or user objects.</li> </ul> <div data-bbox="662 709 1425 835" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>Note</b><br/>The Attachment data type can't be used for <i>externalCode</i> field.</p> </div>   |
| <p><b>Maximum Length</b></p> | <p>The Maximum allowed data length for the data type <i>String</i>. The maximum length includes the total number of digits before the decimal separator for <b>Decimal</b> data type. The default value for this field is 255.</p> <div data-bbox="618 978 1425 1192" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>Caution</b><br/>You aren't allowed to set the field length lower than the length of existing data. If you try do so for external code field or other fields that are part of business key, an error message appears and you aren't allowed to save the change. For other fields, a warning message appears.</p> </div> <p>The <b>decimal separator, thousand separators, +, or -</b> isn't included in the Maximum Length.</p> <div data-bbox="618 1297 1425 1680" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>Note</b><br/>To configure the fields in the Object Definition UI, consider the following limits:</p> <ul style="list-style-type: none"> <li>• Maximum length for <b>externalName</b> and <b>externalCode</b> is 128 characters.</li> <li>• Maximum length for <b>String</b> data type is 4000 characters for predelivered objects.</li> <li>• Maximum length for <b>Number</b> data type is 4000 characters, but it's recommended not to exceed 38.</li> <li>• Sum of Maximum Length and Decimal Precision for <b>Decimal</b> data type is 4000 characters, but it's recommended not to exceed 38.</li> </ul> </div> |
| <p><b>Details</b></p>        | <p>Enter field details.</p>   |
| <p><b>Actions</b></p>        | <p>Delete the field or move its display position up or down.</p>  |

2. Choose *Details* to add additional field attributes.

---

### Valid Values Source

It's mandatory to set this field property for the following data types:

- **Picklist**  
Code of the MDF picklist.
- **Generic Object**  
Object code of the generic object you want to reference.
- **Foundation Object**  
Object type of the foundation object you want to reference. For example, location defined in the Corporate Data Model.
- **Enum**  
Fully qualified class name of the Java enum for this data type.

#### ⚠ Caution

To avoid application errors, such as failure to access quickcards, do not use composite child objects as the value source of a field.

---

### Hide Old Value

Effective dated objects have a history view. In the history view, records for different start dates are shown. A strikethrough is shown to indicate the previous value of the field. If you don't want to see the strikethrough for the field, set *Hide Old Value* to *Yes*.

---

### Decimal Precision

Decimal precision indicates the number of digits after the decimal separator.

#### ℹ Note

Sum of Maximum Length and Decimal Precision for **Decimal** data type is 4000 characters, but it's recommended not to exceed 38.

---

### Include Inactive Users

Set to *Yes* if you want to display data for inactive users, along with active users. Default value is *No*.

---

### UI Field Renderer

Use this field to define a specific UI behavior when selecting a value from a dropdown on Manage Data. Only the specific values configured are shown in the dropdown list.

The possible field values supported by MDF are as follows:

- `displayPickListWithoutExternalCode` (for picklists)
- `displayGOWWithoutExternalCode` (for generic objects)

---

### Transient

Set to *Yes* to make the field transient. Values of transient field aren't stored in the database, but are populated at runtime in the user interface, based on rules or associations.

---

|                            |   |
|----------------------------|---|
|                            | <p>For example, you can use transient fields to display simple calculations at runtime, such as an age or a number of days remaining since a particular date. You can also use transient field to display different text labels based on certain rules, such as an indication of whether a given pay component is shown as a percentage or an amount.</p>   |
| <b>Help Text</b>           | <p>Set this translatable text for the field. You can give instructions for the user.</p> <p>You can format your text with Rich Text Format (RTF) using BB Codes. Using BB Codes you can format your text, add hyperlinks and render them as desired, visually. For information on supported BB Codes, see Related Information.</p>  |
| <b>Mask Value on UI</b>    | <p>Set to <i>Yes</i> or <i>No</i>.</p> <p>If you set this field to <i>Yes</i>, then the value saved on that field is visible as '***'. For example, a password field for any logon page.</p> <p>If you set this field to <i>No</i>, then it behaves like all the other fields.</p> <div data-bbox="841 978 1429 1230" style="border-left: 2px solid #0070C0; padding-left: 10px;"><p><b>Note</b></p><p>This field only handles the display on the UI and doesn't ensure security during data transmission across the network. To ensure that information isn't accessed by an unauthorized user or to enable Read Audit for a field, use the LSPD configuration option.</p></div> |
| <b>Show Trailing Zeros</b> | <p>Set to <i>Yes</i> or <i>No</i>.</p> <div data-bbox="841 1314 1429 1535" style="border-left: 2px solid #0070C0; padding-left: 10px;"><p><b>Note</b></p><p>If you set this field to <i>Yes</i>, then it shows trailing zero for decimal precision. For example, for a decimal field, set the decimal precision to 4, enter value for the field as 12 and save. The value then appears as 12.0000.</p></div>  |
| <b>Default Value</b>       | <p>You can set a default value for the field.</p> <p>When a default value is selected for a field, it initializes the new page with the given value. For example, if you set the default value for a STRING field as ABC, when you go to <a href="#">Manage Data</a> page to create a new object, then you see ABC already listed in that field.</p>  |
| <b>Hide Seconds</b>        | <p>Set to <i>Yes</i> or <i>No</i>.</p>  |

---



---

When you set *Hide Second* to *Yes*, seconds are hidden from the time display.

#### Note

This property is only applicable for Time data type and is ignored for other data types including DateTime.

---

### Required

Set to *Yes* if the field is required.

#### Note

If a field is configured as a Data Subject Field in Legislatively Sensitive Data Configuration for a custom object, then Required property must be set to Yes.

---

### Visibility

Set to one of the following:

- Editable  
Field is visible and editable.
- Read Only  
Field is visible and read only.
- Not Visible  
Field is inactive and won't appear in UI, API, imports/exports, and so on.

#### Note

For fields that have sensitive information, we recommend you to control field visibility using the field overwrite option in role-based permissions.

---

### Status

This field is a read-only field with the value *Active* or *Inactive*. Inactive fields aren't available in the system. You can't use them in import, UI, OData, rules, and so on. If you don't enable the corresponding module, fields become inactive automatically. Note that you can't make the fields inactive from the UI.

---

### Label

Set this translatable label for the field. The default value is the field name. To enter a translated label name, click the translations button.

#### Note

MDF object and field labels can't be modified using

► [Admin Center](#) ► [Text Replacement](#) ►.

---

### Cascade

Set to one of the following:

---

|                        |  |
|------------------------|--|
|                        | <ul style="list-style-type: none"> <li>• No Selection<br/>User sees the field as read only on the <i>Business Configuration UI</i> page.</li> <li>• None<br/>User sees the field as read only on the <i>Business Configuration UI</i> page.</li> <li>• Save<br/>User is able to edit the field on the <i>Business Configuration UI</i> page.</li> </ul>  |
| <b>In Activated By</b> | If this object is dependent on another object, select the object from the list.  |
| <b>End Of Period</b>   | <p>Set this field to <b>Yes</b> in the <a href="#">▶ Object Definition ▶ Details ▶</a> page to enable 24-hour time format in the <i>Manage Data</i> page to distinguish between start date and end date.</p> <p>By default, this field is set to <b>No</b>.</p> <p>This field is useful for Time Off and time sheet scenarios.</p>   |
| <b>Rules</b>           | Select the rule to associate with updating the field. You can also create a new rule to select by using the + button.  |
| <b>Field Criteria</b>  | <p>Set this field to restrict the possible values that it can hold by specifying the following:</p> <ul style="list-style-type: none"> <li>• Source Field Name<br/>Field whose values restrict the possible values for the field that you're defining.</li> <li>• Destination Field Value<br/>Name of the field whose value is being restricted. This is the name of the field that you're defining.<br/>For example, you can restrict choices of dates based on an effective start date, or you can restrict choices such as countries/regions, time zones, and so on, by location.</li> <li>• defaultDestinationValue<br/>You can create a default destination in cases where the destination field value isn't set. Currently it only stores constant values.</li> </ul> <p>You can add or delete field criteria by clicking the corresponding icon.</p> <p>For more information, see <a href="#">Field Criteria [page 45]</a>.</p> |
| <b>Condition</b>       | Used to hide or show the field based on another(parent) field. Enter the parent field ID that determines if this field needs to be hidden or displayed. For composite child fields, this property doesn't support all use cases.   |

---

|                         |   |
|-------------------------|---|
| <b>Condition Values</b> | You can enter a list of values that makes the condition true. Make sure that you enter the values correctly because there are no system validations for this field. |
|-------------------------|---|

3. When you finish configuring field attributes, choose *Done* to go back to object configuration.
4. Follow the same steps to add more fields.

## Related Information

[Supported BB Codes for RTF Formatting \[page 32\]](#)

### 4.2.1.1 Supported BB Codes for RTF Formatting

The following BB codes are supported for errors, warnings and information help texts.

| BB Codes   | Description  |
|--|--|
| <code>[b]</code> , <code>[i]</code> and <code>[u]</code>             | Applies text format such as - bold, italics and underline, respectively.   |
| <code>[color]</code> , <code>[size]</code> and <code>[font]</code>   | Applies preferred fonts, color and size formatting for your help text.   |
| <code>[left]</code> , <code>[right]</code> and <code>[center]</code> | Indents your text towards left, right or center.   |
| <code>[email]</code> and <code>[url]</code>                          | Includes email addresses and links in your help text. xURLs are also supported. xURLs opens the link in a new tab in your browser.   |
| <code>[list]</code>  | Formats your help texts into a list.   |
| <code>[image]</code>   | Inserts images in your help text.  |
|  | <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>Note</b></p> <p>Ensure that the image size is within your configuration limits.</p> </div> |
| <code>[code]</code>  | Inserts example code to your help text.  |
| <code>[quotes]</code>  | Inserts quotes in your help texts.   |
| <code>[noparse]</code>   | Retains BB codes in your help text by using this code.   |
| <code>[sub]</code> and <code>[sup]</code>                            | Applies subscript and superscript text to your help texts.   |
| <code>[br]</code>  | Inserts line breaks in your help text.   |

### Note

- When you modify messages in a rule, ensure that you have saved your changes.
- We recommend that you test the readability of your RTF formatted help text before you enable it for the output.

## 4.2.2 Adding Associations

Adding associations to an object allows you to define a relationship between objects. These objects can either be generic object or a foundation object

### Context

As an Admin, you can add association to an object definition using **Configure Object Definitions** page.

### Procedure

1. Go to [Admin Center](#) > [Configure Object Definitions](#).
2. From the *Search* dropdown, select **Object Definition**.

The second dropdown gets populated with values corresponding to the selected object.

3. Select a object definition from second dropdown.

UI displays all the configurations corresponding to the selected object definition.

4. Click [Take Action](#) > [Make Correction](#).
5. In *Associations* section, click *Details*.

*Details* UI appears.

6. In the *Name* field, provide unique name for the association.
7. From the *Multiplicity* dropdown, select a Multiplicity type.
8. From the *Destination Object* dropdown, select the object that you want to associate with.
9. From the *Type* dropdown, select the association type.
10. In *Destination Object Column*, provide the field name of the child object to which the object should be mapped.
11. In *Source Object Column*, provide field name of the parent object to which the object should be mapped.
12. From the *isTransient* dropdown, select a value.

### Note

By default this field is set to **No**. To make a field transient, set the field to **Yes**. If a field is defined as transient, values of the field are not stored in the database, but are populated at runtime in the user interface, based on rules or associations.

13. In *UI Association Renderer* field, define the layout of Composite Association in a data grid layout for repeating forms.
14. From the *Required* dropdown, select **Yes** if association is required for object definition else set this field to **No**.

#### ⓘ Note

If required, at least one record for the association must exist when an instance is created of the object that you are defining.

15. From the *Visibility* dropdown, select the Visibility type.

#### ⓘ Note

This field can have the following values:

Editable: User can add and delete data for this association.

Read Only: User can only view records of this association. A user cannot add, delete, or edit any association data.

Not Visible: User cannot view this association.

16. From the *Status* dropdown, select status.

#### ⓘ Note

This is a read-only field. Possible values are Active and Inactive. Inactive fields are not available in the system. You cannot use them in import, UI, OData, rules, and so on. If you do not enable the corresponding module, fields become inactive automatically. Note that you cannot make the fields inactive from the UI.

17. In the *Label* field, provide label for the association. Here you can also add translated labels.
18. From the *Cascade* dropdown, select the cascade type.

#### ⓘ Note

You can choose any of the following:

Set to one of the following:

No Selection: User will see the field as read only on the Business Configuration UI page.

None: User will see the field as read only on the Business Configuration UI page.

Save: User will be able to edit the field on the Business Configuration UI page.

19. From the *Inactivated By* dropdown, select the object on which the current object is dependent, if any.
20. From the *Allow Reordering* dropdown, select **Yes**, to allow reordering of associations on **Manage Data** page else set this field to **No**.

#### ⓘ Note

By default, this is set to **Yes**. When you set this to **No**, the system automatically sorts the association data based on the last modified date-time stamp.

21. In *Field Criteria*, specify *Source Field Name*, *Destination Field Name*, *Default Destination Value* to restrict the possible values for the field.

#### Note

*Field Criteria* is valid only for **valid-when** associations.

22. In *Condition* field, provide the field ID.

23. In *Condition Values* field, provide the condition value.

#### Note

You can enter a list of values that makes the condition true. Make sure that you enter the values correctly because there are no system validations for this.

## 4.2.3 Adding Workflows

This section describes how to add workflow configuration to the object that you are defining.

Workflows are approval processes set up for another user (typically a manager) to approve or reject data changes. In a typical workflow scenario, if the user submits a data change, the approver gets an approval request in the To Do list. The change is not processed by the system until the approver approves the request. It is a good idea to set up workflow routing for any data changes that need to be reviewed by another user. In the object definition, the workflow will appear in the respective To Do section. Once the workflow is approved, it is removed from the user's to do list.

You can add workflow routing to an object definition by associating it with a workflow configuration. Use workflow routing if you want any data changes to the instance of the MDF object that you are defining to trigger an approval workflow. Workflows can be used to fulfill business use cases such as leave approvals, candidate hiring processes, and salary raises. From the dropdown of the available workflow routing, choose the one that you want to associate with the object that you are defining.

#### Note

To create new workflow configurations to add to the dropdown, navigate to the *Admin Center*. In the *Tools* search field, enter *Manage Organization, Pay and Job Structures*, then select *Workflows*. For more information, see the *Employee Central Workflows* guide from the SAP Help Portal.

The new workflow configuration that is created is automatically available in the dropdown menu for object definition.

Here is a description of the supported workflow actions:

| Workflow Action | Description   |
|-----------------|---|
| Approve         | The change is approved. The record is processed and the change appears in the system.     |
| Decline         | The change is declined. The request is not processed and no change appears in the system. |
| Withdraw        | A request is withdrawn. No change appears in the system.                                  |

| Workflow Action          | Description  |
|--------------------------|--|
| Request                  | A request is submitted and sent for approval.  |
| Comment                  | Workflow participants add a comment to the request, which can be read by other participants.   |
| In-flight change         | Approver makes minor changes to a request and then approves it. For example, an employee claims an insurance policy premium as part of IT Declarations and includes the late payment fee, which is not eligible. The administrator can approve only the basic insurance policy premium and edit the approved amount accordingly. |
| Send back (or push back) | Approver sends back a request so that the requestor can edit and re-submit the same request. This enables multiple use cases for IT Declarations, Reimbursements and Advances functionality.   |

### Note

The transient fields, for example **Pay Range**, which stores the dynamic data in the [Workflow Details](#) page, display values when the page is loaded. This is particularly useful for HR Admins as they can see all the data required to approve or decline the change requests.

For more information on transient field, see [Adding Fields \[page 25\]](#).

## 4.2.4 Adding Searchable Fields

This section describes how to add searchable fields to an object dDefinition.

### Context

You can define fields that can be used to search for an MDF object. Search fields can be used in the search panel on the [Manage Data](#) page to search for objects.

### Caution

The search result doesn't respect field-level permissions. If you have a sensitive field that must be visible only for a few specific people, don't make it searchable.

Searchable Fields

\* Field                      \* status

Click or focus to edit      Active

## Behavior of Search Field During Auto-Complete

Consider the scenario where a Generic Object O1 has composite children of type OC1. OC1 has two fields, F1 and F2. The children.F1 and children.F2 are two searchable fields for Generic Object O1. There is an instance P1 of O1, which has two composite children instances, C1 and C2.

During autocomplete on composite children of type OC1, if the input string matches the field F1 of C1 and field F2 of C2, auto-complete query returns P1 twice. If fields F1 and F2 have same values for both C1 and C2, P1 is returned twice.

## Procedure

1. Go to [Admin Center](#) > [Configure Object Definitions](#).
2. From the [Search](#) dropdown, choose [Object Definition](#).
3. Search for and select the object you want to configure, and choose [Take Action](#) > [Make Correction](#).
4. In the [Searchable Fields](#) section, enter the names of the fields and/or association that you want to make searchable. You can add more fields and associations by pressing  after each field name.
5. Save your changes.

## 4.2.5 Adding Business Key Fields

This section describes how to add business key fields to an object definition.

Business key fields are a combination of two or more fields and are defined for an object so that you can uniquely identify its records.

For example, if the business requirement is to ensure an employee is assigned a piece of equipment and the same equipment and employee combination must not be repeated in the system. To meet this requirement, we can set a combination of UserID and EquipmentID fields of the EquipementAssigement object as the business key fields. However, you must ensure that all the fields in the business key combination are fields of the same object. Fields from associations can't be part of business key fields.

### Note

External Code must always be unique even if the business key is defined.

To maintain the uniqueness of records automatically, we recommend that you configure External Code for such an object with the data type as Auto-number.

#### Business Key Fields

Field

In the field, enter the business key fields. You can delete or add more fields by clicking the corresponding icon.



## 4.2.6 Adding Security

This section describes how to add security to the MDF object definition.

SAP recommends that all custom MDF objects are created as RBP secured. Use the Security configuration page to define and add security at both object level and field level.

For information about adding object, field, and association level security to the MDF object that you're defining, see [MDF Security Using Role-Based Permissions \(RBP\)](#).

| Field                                  | Directions  |
|--|---|
| <i>Secured</i>                         | <ul style="list-style-type: none"><li>For custom objects, this field is set to <i>Yes</i> by default when the <i>Effective Dating</i> in the <i>Properties</i> page is set to <i>None</i>, <i>Basic</i>, or <i>Multiple Changes Per Day</i>.</li><li>For custom objects, this field is by default set <i>No</i> when the <i>Effective Dating</i> in the <i>Properties</i> page is set to <i>From Parent</i>. You can't configure object-level security settings of this type of object definition. If you want to configure field level overrides, choose <i>Yes</i>.</li><li>For predelivered objects, you are only allowed to change the configuration of this field for a few of them under certain scenarios.</li></ul> <div data-bbox="818 1031 1427 1184"><p><b>Note</b></p><p>Don't select <i>Custom</i>. The option isn't supported and will be removed in future releases.</p></div> |
| <i>Permission Category</i>             | Select a permission category where you can find the object definition later in the <a href="#">Manage Permission Roles</a> tool.  |
| <i>RBP Subject User Field</i>          | Specify a required user field of the current object definition. The value of this field is used to set up target population in RBP.   |
| <i>CREATE Respects Target Criteria</i> | <ul style="list-style-type: none"><li>When this field is set to <i>Yes</i>, users can only create new objects whose values of fields are within their target population.</li><li>When this field is set to <i>No</i>, users can create new objects regardless, but don't have access to those objects beyond their permissions.</li></ul>   |

| Field  | Directions  |
|--|---|
| <a href="#">Base Data Field For Blocking</a> | <p>Specify a required date field of the current object definition. The value of this field is used as the effective end date for data blocking in RBP.</p> <ul style="list-style-type: none"> <li>When the <a href="#">Effective Dating</a> field is set to <i>None</i>, specify a required date field. If this field is left blank, you can't set up data blocking for this object definition in RBP.</li> <li>When the <a href="#">Effective Dating</a> field is set to <i>Basic</i> or <i>Multiple Changes Per Day</i>, specify a required date field of the current object definition. <ul style="list-style-type: none"> <li>When a user field is specified in the <a href="#">RBP Subject User Field</a> and this field is left blank, this field is set to <a href="#">mdfSystemEffectiveEndDate</a> automatically after saving.</li> <li>When both <a href="#">RBP Subject User Field</a> and this field are left blank, you can't set up data blocking for this object definition in RBP.</li> </ul> </li> </ul> |

After an object is secured, you can control access to that object in role-based permissions.

## Related Information

[Setting Up Security for Composite Child Permissions Data Blocking](#)

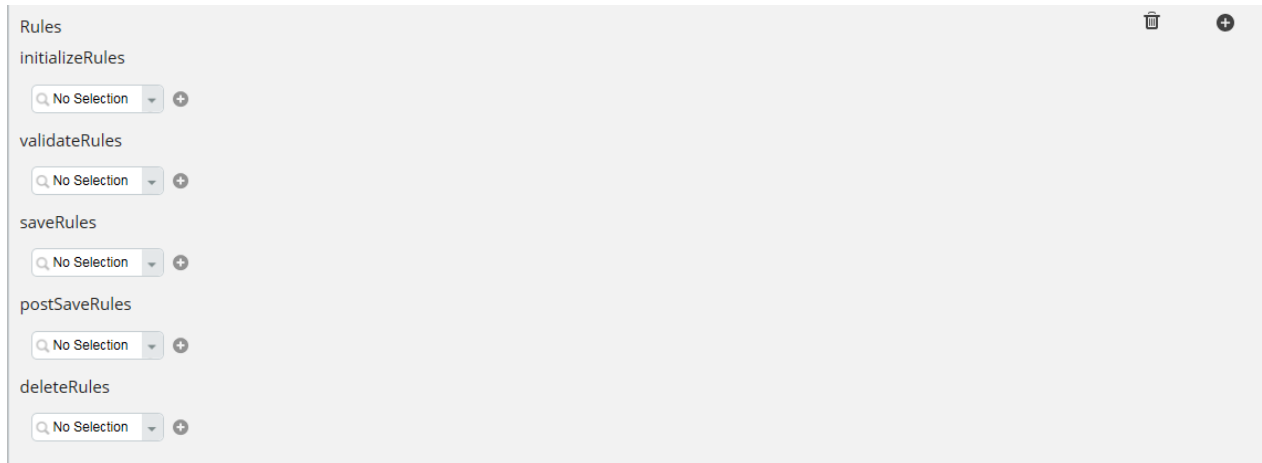
### 4.2.7 Adding Rules

This section describes how to add rules to the object that you're defining.

Rules are a preset way to add application logic to determine the outcome of a change made to particular data in the system. You can add custom business logic to an object definition by associating it with one or more rules. Rules are triggered by different actions by a user on data. For example, you want to set a rule such that data changes trigger an approval request from a manager or administrator, and that the changes are rejected if not approved.

Rules are a way to maintain a consistent application logic on the occurrence of an event. They can also be a way to keep a check on data changes (by triggering an approval). Rules can also be used to autofill field data (for example, time zone information can be autofilled based on location information).

You can't edit any rules from this UI. You can only edit rules in the [Configure Business Rules](#) UI.



## Rule Categories

| Rule Category   | Description  |
|-----------------|--|
| initializeRules | <p>Initialization rules ("initializeRules") are triggered in the initialization phase of object creation (an "onInit" event). You can use these rules to initialize fields to default values. OnInit is basically a constructor and no context parameters are provided by onInit.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>Initialization rules aren't supported when setting the wfConfig attribute to trigger workflows.</p> </div> |
| validateRules   | <p>Validation rules ("validateRules") are triggered after a change to an object is submitted but before the change is saved (an "onValidate" event). You can use these rules to validate field values.</p>   |
| saveRules       | <p>Save rules ("saveRules") are triggered when a user tries to save changes to an object (an "onSave" event). You can use these rules to check related field values for correctness. For example, a field could become required as a consequence of a save.</p>  |

| Rule Category | Description   |
|---------------|---|
| postSaveRules | <p>Post-save rules ("postSaveRules") are triggered after changes to an object have been saved (an "onPostSave" event). These rules are used when you want to send an alert message to the user. They aren't used to set a field value. To activate the alert and notification feature, you must associate them with a workflow.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>By the time a post-save rule is triggered the MDF object has already been saved and can't be changed. Also, post-save rules aren't supported when setting the wfConfig attribute to trigger workflows.</p> </div> |
| deleteRules   | <p>Deletion rules ("deleteRules") are triggered after an object record is deleted (an "onDelete" event).</p>  |

## Rule Hooks

| Rule Category | Rule Hook/Event Point   |
|---------------|---|
| onInit        | <p>No previous, original, or future Generic Object is passed on to the rule.</p>  |
| onValidate    | <p>Previous Generic Object, original Generic Object, and future Generic Object are passed on to the rule for parent objects.</p> <p>This doesn't include composites. However, rules attached to the composite child are also triggered.</p> |
| onSave        | <p>Previous Generic Object, original Generic Object, and future Generic Object are passed on to the rule for parent objects.</p> <p>This doesn't include composites. However, rules attached to the composite child are also triggered.</p> |
| onPostSave    | <p>Previous Generic Object, original Generic Object, and future Generic Object are passed on to the rule for parent objects.</p>  |
| onChange      | <p>Previous Generic Object, original Generic Object, and future Generic Object are passed on to the rule for parent objects.</p> <p>This doesn't include composites.</p>  |

| Rule Category | Rule Hook/Event Point   |
|---------------|---|
| onDelete      | <p>Previous Generic Object, original Generic Object, and future Generic Object are passed on to the rule for parent objects.</p> <p>This doesn't include composites. Rules attached to the composite child <b>aren't</b> triggered.</p> |
| onLoad        | <p>It is for loading the object and thus no previous, original, or future Generic Object are passed on to the rule.</p>   |

## Need to Know

- You can define rule functions that accept the Time/DateTime MDF data type as parameters and as return type.
- Change rules ("onChangeRules ") are used at the field level. Objects can't be associated with change rules. Change rules are triggered when the value of a particular field is changed. You can use these rules to populate another field after the change. For example, you can populate the Country field with a certain value based on a change to the Position field.  
For fields, rules are triggered only when a field change event is triggered in the user interface, from the [Manage Data](#) and [Manage Configuration UI](#) tools. OnChange rules aren't triggered by the ODATA API or by data import. OnChange events are triggered on INSERT\_NEW apart from CREATE and CORRECTION from UI. OnChange event is triggered when user clicks on INSERT\_NEW and selects the date. The user can attach the rule on the effective start date and that rule is triggered for INSERT\_NEW. You can use field level rules to automatically set the values of other fields based on the value selected for the current field.
- You can also trigger business rules after an onLoad event in the UI. This enables pages such as the [Manage Configuration UI](#) and [Workflow Approval](#) to trigger the calculation of transient fields and display the results in the UI.
- MDF also passes the operation (such as INSERT/CORRECTION/CREATE) to the rule engine in the context (if it's available to the MDF).
- For the new object definition, the [Configure Object Definition](#) page lists only those rules that are valid to that object definition.  
For an existing object definition, you can't add invalid rules. All the existing rules, even though they're invalid, work as before. Whenever you update an object definition with invalid rules, a warning message is displayed to either remove the rule or correct it.

## 4.2.8 Adding Translation Labels for Country/Region and Currency

Translated labels allow you to display country/region and currency labels in a localized language.

### Context

As an Admin, you can add country/region-specific translations for a label using **Manage Data** page. If no translation is maintained for a country/region, the **Default Value** is displayed on the UI.

### Procedure

1. Go to [Admin Center](#) > [Manage Data](#).
2. From the **Search** dropdown, select an object definition. For example, **Country/Region**.  
The second dropdown gets populated with values corresponding to the selected object.
3. Select a value from second dropdown. For example, **Argentina (ARG)**.
4. To edit the object, select [Take Action](#) > [Make Correction](#).
5. To add translations for an object label, select the icon next to the object label field. If you selected **Currency** as the object value, you see the icon next to the *Currency* field. Likewise, if you selected **Country/Region**, you see the icon next to the *Country/Region* field.

The **Translations** UI appears. Here, you can add or modify translations.

6. Provide country/region-specific translations.
7. Select **Done**.

You're taken to **Manage Data** page.

### Next Steps

It's good to validate the changes you made in the UI, as the user would see it.

## 4.2.9 Creating MDF Blocks

You can create customized MDF blocks on entities built using conditional associations.

#### Note

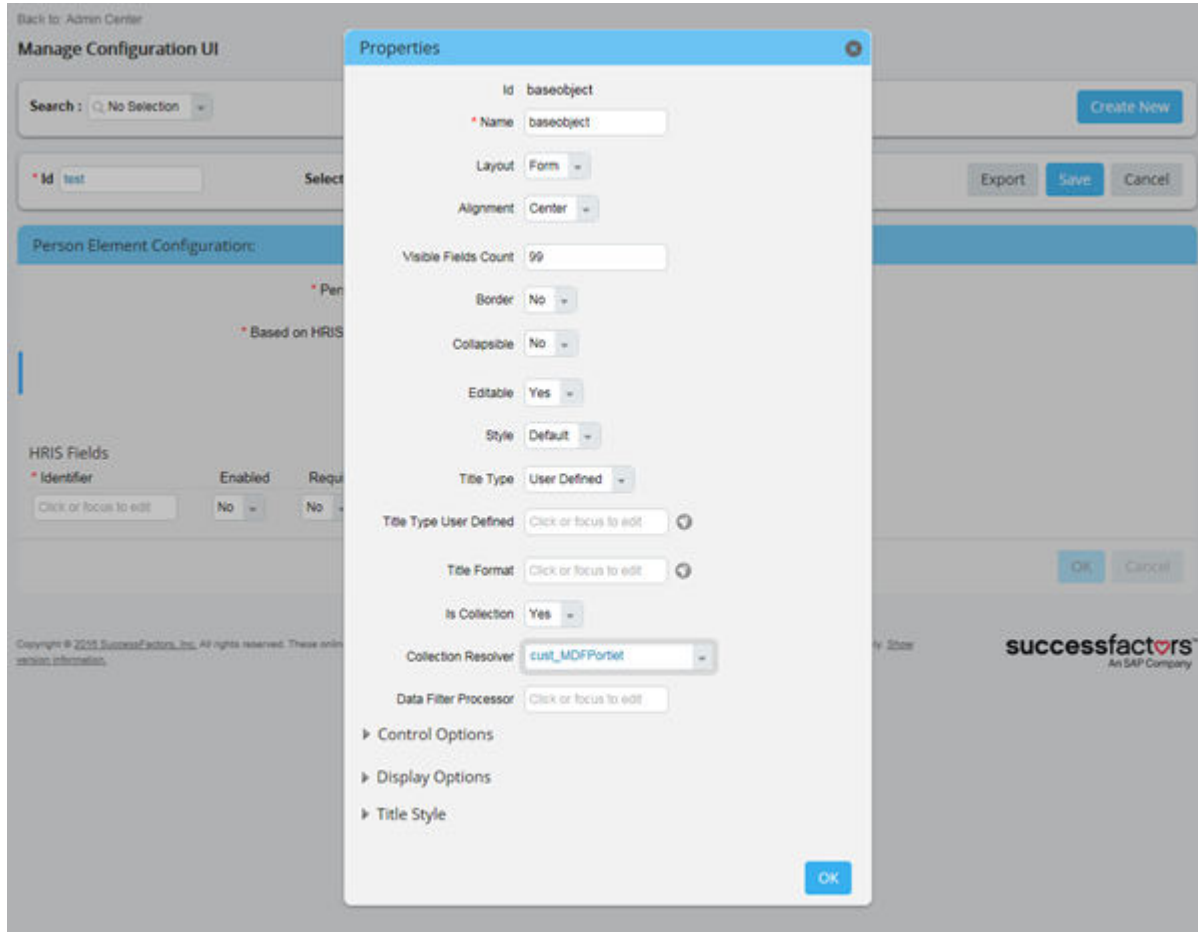
Conditional associations are also known as country/region-specific fields.

| Fields                  |                       |                        |             |           |
|-------------------------|-----------------------|------------------------|-------------|-----------|
| * Name                  | * Database Field Name | Maximum Length         | * Data Type | (20) More |
| externalCode            | externalCode          | 255                    | String      | Details   |
| externalName            | externalName          | 255                    | String      | Details   |
| cust_effectiveStartDate |                       | 255                    | String      | Details   |
| cust_dr1                |                       | Click or focus to edit | String      | Details   |
| cust_dr2                |                       | Click or focus to edit | String      | Details   |
| cust_                   |                       | Click or focus to edit | String      | Details   |

| Associations |                |                             |           |           |
|--------------|----------------|-----------------------------|-----------|-----------|
| * Name       | * Multiplicity | * Destination Object        | * Type    | (12) More |
| cust_as1     | One To One     | Association                 | Composite | Details   |
| cust_as2     | One To One     | Association                 | Composite | Details   |
| cust_as3     | One To One     | Accrual Lookup By Senior... | Composite | Details   |
| cust_        | One To One     | Absence Counting Method     | Composite | Details   |

You can use the configurable UI to create these MDF blocks and configure the UI for conditional association on the *Manage Configuration UI* page. Once you've entered the *ID* and *Base Object*, in the *Configuration* section, select *Edit Properties*. You can also configure properties such as *Layout*, *Visible Fields Count*, *Border*, and *Collapsible*. And change the group layout and move the group within the UI.



### Note

When you create, edit, or delete a record on a custom MDF block with a workflow attached to it, a link is displayed on the block with the pending record and the effective date. When you click the link, you can only view the records and can't create or edit any records until the request is approved.

## 4.3 Understanding Object Definitions and Relationships

To use object definition it is important to understand important concepts like field criteria, associations in MDF, and auto-number ExternalCode.

### 4.3.1 Field Criteria

This section describes field criteria and how to define a field criteria based on your business use case.

Field Criteria is a configuration defined at a field or a valid-when association level, that allows customer to restrict values of the field based on value of another field.



| Attribute                 | Description  |
|---------------------------|--|
| Source Field Name         | <p>The field value that you want to control by applying restrictions. For example, if the field is of type FO and the object is legal entity, this attribute can be the field name in the legal entity for which you want to control the output, for example, Startdate. This is a required field.</p> <p>This attribute is a field on the referenced object that is restricted or filtered.</p> <p>For example, if the field has the data type Generic Object and the valid value source is LegalEntity, and startDate is a field on the object LegalEntity. In this scenario, the field criteria will filter the available values for Legal Entities where LegalEntity.StartDate matches the filter condition, which matches the value derived from the <a href="#">Destination Field Name</a> or <a href="#">Default Value</a> fields.</p> <p>You must provide a value for this field when defining field criteria.</p> |
| Destination Field Name    | <p>This attribute represents a field name, either on the same object or on any of its parent objects in the hierarchy.</p> <p>The value of this field is used to restrict or filter the values to which the restriction is applied.</p>  |
| Default Destination Value | <p>This attribute stores a constant value.</p> <p>The attribute value is used if the <a href="#">Destination Field Name</a> is not provided or is empty.</p>   |

### 4.3.1.1 Validating Field Criteria

This section explains the validation of field criteria.

For the field criteria to be valid, the client must pass the following:

- Source field name
- Destination field value or the default value of the destination field or both

The system validates whether you have entered a valid field name.

## 4.3.1.2 Limitations

This section describes the limitations of field criteria in MDF.

The following limitations apply to selection of field criteria in MDF:

- If there are multiple criteria only the AND condition works on the resultant query. The OR condition cannot be applied.
- Field criteria related to a date field can only be applied using the standard field `effectiveStartDate` field criteria to show that the field values based on any other date field is not supported.
- Dates can only be entered in the YYYY-MM-DD format.
- Default value of a field has to be a constant. It cannot be obtained from a different object or field.
- For Foundation Objects, MDF does not validate the field name. It just stores the value as passed. However, an invalid field name results in a runtime error when you apply the field criteria.

## 4.3.1.3 Filtering Using Field Criteria

This section shows how to filter using field criteria.

You can filter object content based on related MDF objects and foundation objects. You can use field criteria to filter using the attributes of a foundation object.

## 4.3.2 Associations in MDF

Associations define a hierarchical relationship between two objects. You can add an association to define either a parent-child relationship or a valid-when relationship between the object definition that you are creating and another object. The other object can either be any generic object or a foundation object.

As an Admin, you can add child objects with one-to-many (OTM) or one-to-one (OTO) relationship to the object that you are defining.

### ❁ Example

a business unit consists of several departments so you create an association of one business unit to many departments; this is a ONE\_TO\_MANY relationship. A location can only have one geozone associated with it; this is a ONE\_TO\_ONE association.

For MDF, the visibility of an object within an association depends on the security setting of both objects. For example, if a child object has security restrictions on fields or the entire object, those restrictions will determine what is visible to users when they access that association.

MDF currently supports associations between the following objects:

- Generic object to generic object.
- Foundation object to generic object.

#### Note

This association is defined in the corporate data model. For more information, see the [Employee Central guide](#).

## 4.3.2.1 Association Types

Associations provide a way to relate objects that can be grouped together logically. The type of association restricts what the user can display or enter. For example, if you set up a ONE\_TO\_ONE association from location to geozone, the user can enter exactly one geozone for a location on the UI.

MDF supports the following association types:

- **Valid When**

For this association type, the associated object has its own lifecycle and exists even without the object that is being defined.

For example, a legal entity object can have a valid-when relationship with a location object. Using that association, when you add a legal entity, you are shown an autocomplete for the available locations that are active and effective on the current entity's start date. If the current entity is non-effective-dated, the results for today's date (according to the system) are shown.

- **Composite**

For this association type, the entity being associated is a child entity that does not exist outside the parent object that you are defining. If an object is referred as a composite child in an object, it cannot be used as a top-level object. It is not shown in the dropdown when you create a new object.

For example, a location can have a composite association with an address object or a picklist with picklist values. An address object cannot exist without a location object. When you enter location details, you see a data grid with address fields.

If you want to make a composite association, the destination object must have the FROM\_PARENT effective dating. Note that an object that is used as a composite object cannot be reused as a valid-when object because it exists within its parent only.

#### Caution

Make sure you follow the number limit of composite children you can create for an object in [Important Configuration Guidelines and Usage Limits \[page 10\]](#).

- **Join By Column**

You can use this association type to model associations that are neither composite nor reference (Valid When) associations. Set one field of the parent object definition as the source object column and one field of the child object definition as the destination object column, then join-by-column association will make the value of these two object columns reference each other.

For example, you have created two entities, Person, which is non-effective-dated, and PersonInfo, which is effective-dated. Person is the parent object and the externalCode column is used as the source. PersonInfo is the child object and, using join-by-column, its externalCode column refers to the Person object. In this way, these two objects always share the same externalCode.

#### Note

Currently, Join By Column association type is not supported for Reporting.

#### 🕒 Note

You can create this type of association implicitly in the source, if the field in the target contains 'inverse = true' to avoid circular dependency.

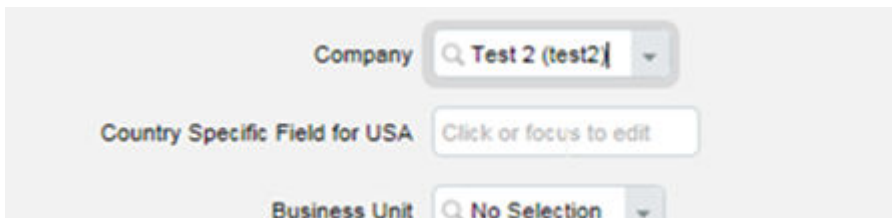
#### ⚠️ Restriction

You are not allowed to define Join By Column associations for custom MDF objects. Any existing Join By Column associations for objects will continue to be honored and are not impacted.

### 4.3.2.2 Example of Country/Region-Specific Conditional Fields in Position Management

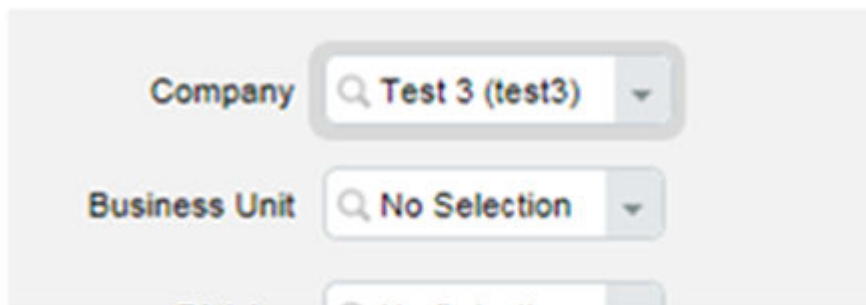
You want to create a country/region-specific field that is displayed only when a certain value is entered in the *Company* field when creating a position.

1. Create the field `cust_countryspecificfield1`.
2. Make the field dependent on the company (FO object that is part of the position definition) and choose the label **Country Specific Field for USA**.
3. Add the following condition values: Test 1, Test 2.
4. When creating a position, choose Test 1 or Test 2 as the company. The country/region-specific field is displayed.



The screenshot shows a form with three fields. The 'Company' field is a dropdown menu with 'Test 2 (test2)' selected. Below it, the 'Country Specific Field for USA' field is visible and contains the text 'Click or focus to edit'. The 'Business Unit' field is a dropdown menu with 'No Selection' selected.

If you change the company to something other than Test 1 or Test 2, the country-specific field is no longer displayed.



The screenshot shows a form with two fields. The 'Company' field is a dropdown menu with 'Test 3 (test3)' selected. Below it, the 'Business Unit' field is a dropdown menu with 'No Selection' selected. The 'Country Specific Field for USA' field is not visible, indicating it is hidden.

### 4.3.2.3 Conditional Associations and Fields

This section describes how to add conditional associations to an object definition.

You can define conditional associations for their MDF objects. If you want an association that is not always valid and depends on a parent field value, you use a conditional association. (Conditional associations are also known as country/region-specific fields.)

For example, address data structure can vary from country to country and is not always valid. For this object, you can define a conditional association.

You can enable or disable a conditional association (such as composite or valid-when) based on the specific values of a parent field. For example, Job Info has composite associations such as Job Info USA, Job Info India, and so on. If the country value is USA and India, you can display the associations Job Info USA and Job Info India respectively. This aspect depends on the country field value of Job Info.

Similarly, you can enable or disable a conditional field based on the field value available at the same level. To make a field or association conditional, you must add a condition when defining the field or association while configuring your object definition.

#### Note

The conditional association or conditional field is displayed after the Condition Field ID field. So you must ensure that Condition Field is the last field. For example, if Job Info has Country as the third field and JobInfoIndia as the conditional association using that field, once you choose the country as India, the Job Info India association is displayed after the Country field. So you must change the order of Job Info fields in such a way that Country is the last field.

### Condition Field ID

In this field, you can enter a parent field name or dot-notated field name. For example, if Job Info has a field named Company that refers to the generic object (GO) Company, which in turn has a field named Country that refers to Country GO, you can enter the field ID as `company.country.code`. You must refer to a field that is available in the parent or to a field in association. The field ID in condition cannot refer to field of data types.

For a GO field, even if you want to refer to GO *externalCode*, you must mention the field name of the external code. For example, if Job Info has GO field *Country* referring to Country GO, you must mention *country.externalCode* in the *Condition Field ID* field, where the *externalCode* is the field name inside Country GO.

For a foundation object (FO) field, you can only refer to the FO external code or name. For example, if Job Info has a field with the name *jobCode* that refers to an FO, you can mention *jobCode.externalCode* or *jobCode.name*.

For a picklist field, you can enter the picklist field ID and picklist value as the condition value. For example, if *maritalStatus* is a field name in Job Info, which refers to a picklist relationship status that has the values *married* and *single*, you must give the condition field ID as *maritalStatus* and the value as either *married* or *single*.

## Condition Values

This is a list of values that makes the condition true. It is a free text and you must enter the values correctly because no validations are available.

If there are multiple values, they are treated as an OR condition. For example, if you enter the condition value USA or IND, the condition is satisfied if the value is either USA or IND.

| Condition Field Data Type   | Condition Field ID  | Condition Values                       |
|---|---|--|
| Generic Object (GO)   | Field name available in the referred GO or dot-notated field name | Valid value as per the chosen field ID |
| Foundation Object (FO)  | External code or name   | Valid FO external code or name         |
| Picklist  | Field name  | Valid picklist value                   |
| Boolean   | Field name  | True or false                          |
| Enum  | Field name  | A or I                                 |
| String/Number/Decimal   | Field name  | Valid field value                      |
| AUTO_NUMBER/DATE/ LOCAL-<br>IZED/DATA_SOURCE/ ATTACH-<br>MENT/USER/CLOB | Not allowed   | Not allowed                            |

## Permissions

You can apply role-based permissions (RBP) for conditional fields and associations.

## Limitations

If the association has a conditional field or conditional association, the field on which the conditional association is set and the dependent field must be positioned at the end.

## 4.3.3 List of MDF Core Objects

MDF core objects are objects that consist the Metadata Framework and its features. This topic lists all MDF core objects in the system.

### ⚠ Caution

You must not change the configuration of MDF core objects except for the object labels. All MDF core objects are non-RBP secured. Do not change the security attribute of these objects.

| Object Code                             | Default Label                              |
|---|--|
| AttachmentFieldConfig                   | Attachment Field Configuration             |
| BusinessKeyField                        | BusinessKeyField                           |
| CollectionInfo                          | CollectionInfo                             |
| Condition                               | Condition                                  |
| ConditionValue                          | ConditionValue                             |
| ConfigUIMeta                            | ConfigUIMeta                               |
| ContextField                            | ContextField                               |
| CriteriaBasedHierarchy                  | CriteriaBasedHierarchy                     |
| CriteriaBasedHierarchyAssociationDetail | CriteriaBasedHierarchyAssociationDetail    |
| CriteriaBasedHierarchyDetail            | CriteriaBasedHierarchyDetail               |
| ExtensionCenterPublishStatus            | ExtensionCenterPublishStatus               |
| FieldInfo                               | FieldInfo                                  |
| GDPRConfig                              | GDPRConfig                                 |
| GOAssociationDefinition                 | Association                                |
| GOFieldCriteria                         | GOFieldCriteria                            |
| GOFieldDefinition                       | Field                                      |
| GOLocalizedData                         | Translations                               |
| GOObjectDefinition                      | Object Definition                          |
| GOObjectDefinitionReference             | GOObjectDefinitionReference                |
| GOParentObject                          | GOParentObject                             |
| GOPickListReference                     | GOPickListReference                        |
| GORBPCConfig                            | Security                                   |
| GORuleConfig                            | Object Rules                               |
| GOSimpleExtension                       | Simple Extension                           |
| JSONMeta                                | JSONMeta                                   |
| LSPDConfig                              | Legislatively Sensitive Data Configuration |
| MDFAccessPeriodCriteria                 | MDFAccessPeriodCriteria                    |

| Object Code                                  | Default Label                                |
|--|--|
| MDFAccessPeriodCriteriaDetail                | MDFAccessPeriodCriteriaDetail                |
| MDFAccessPeriodCriteriaMap                   | MDFAccessPeriodCriteriaMap                   |
| MDFAlertType                                 | MDFAlertType                                 |
| MDFCustomEntitiesWithMultipleParentsRegistry | MDFCustomEntitiesWithMultipleParentsRegistry |
| MDFSynchRecord                               | MDFSynchRecord                               |
| MDFTargetCriteria                            | Target Criteria                              |
| MDFTargetCriteriaDetail                      | Target Criteria Detail                       |
| MDFTargetCriteriaInValue                     | MDFTargetCriteriaInValue                     |
| MDFTargetCriteriaMap                         | Target Criteria Map                          |
| MDFTargetCriteriaTypedHierarchyDetails       | MDFTargetCriteriaTypedHierarchyDetails       |
| ObjDefDependencyIndex                        | ObjDefDependencyIndex                        |
| ObjectConfig                                 | Object Configuration                         |
| ObjectDefinitionStatus                       | ObjectDefinitionStatus                       |
| PickList                                     | PickList                                     |
| PickListValue                                | Picklist Value                               |
| RBPFieldOverride                             | RBPFieldOverride                             |
| RBPObjPermission                             | Object Permission                            |
| RBPRoleObjPermissionMap                      | Object Permission Map                        |
| RDPConfig                                    | RDPConfig                                    |
| ReferenceInfo                                | ReferenceInfo                                |
| Rule   | Rule   |
| RuleReference                                | Rule Reference                               |
| SearchableField                              | Searchable Field                             |
| SimpleFieldAndValue                          | SimpleFieldAndValue                          |

### 4.3.4 Auto-Number ExternalCode for MDF Objects

MDF Objects require an external code that acts as a generic key field for an object instance. If more than one key field is required to represent an instance, you can configure the external code field to be invisible and of the data type **Auto-Number**. The auto-numbers are populated from a stored sequence and the sequence is shared by all the Auto-Number fields.

Previously, the Business Key was not defined for many objects with Auto-Number external code. Without a Business Key, instances can only be updated by external code. So, you were allowed to provide the auto-number external code in Imports and OData API, although the field was invisible. Thus, instances were created with the given number without increasing the Auto-Number sequence or updated instances that already existed in the MDF. Whenever you try to create a new instance using the sequence, it was possible that you could come across duplicate key exceptions.



## Temporary Numbers in Imports

If an object does not support a Business Key, temporary numbers can be provided in Imports. The temporary numbers are preceded with # and are replaced with numbers from the sequence when the object is saved. To update an object through import, the exact number must be known and can be provided directly.

## Business Keys in Imports

We recommend to do the import using business key. The business key fields are used to identify the existing objects or create new objects.

### 4.3.4.1 Limitations for Instance Sync for MDF Export and Import

Instance Sync will not work reliably for objects with Auto-Number external code.

Previously, Instance Sync used the Auto-Number gap and copies the numbers from the source to the target instance, which can lead to data corruption. Also, Instance Sync could override objects that were manually created in the target system or it can create objects with Auto-Numbers that exceed the high value of the sequence in the target system, which can lead to duplicate key exceptions at a later point in time.

If the external code is not stable between two systems, the references can break or point to different objects after an instance sync.

### 4.3.4.2 Methods to Avoid Using Auto-Numbers

There are some alternatives with which you can avoid using Auto-Numbers.

- Generating a Semantic ExternalCode
- Using a constant default value in One-to-one Composite Association

## Generating a Semantic ExternalCode

An object, for example, Region is supposed to represent the organizational substructure of a country such as the states in the US, the provinces in Canada or the departments in France. In this case, the object will have two business key fields, **country** and **regioncode**. When you want to concatenate these two fields to an ExternalCode, the result may look similar to USA-CA for California or FRA-75 for Paris.

In such approach, it is required to consider the following aspects while implementing the processor:

- The fields used for concatenation should be mandatory fields.

- The ExternalCode should be read-only.
- You should not change the field after the external code is assigned.

## One-to-one Composite Association

From business perspective, no ExternalCode is required for such associations, but technically MDF requires it. We recommend to define a non-editable external code field as **Number** or **String** and define a default-value for the field. In this case the frame-work will take care of populating the field with a constant value taken from the default.

### 4.3.4.3 Need to Know

This section lists some important points that you should be aware of while using Auto-Number ExternalCode for MDF Objects.

- OData API does not support Business Keys. You can upsert instances with Auto-Number external code with null value. This is interpreted as 'Create' and the numbers will be set from the sequence. With the validation, OData API returns an error. If the external code is set with a number that corresponds to an existing instance, the instance will be updated.
- When the objects are more likely to be copied and updated between different systems:
  - Configuration Objects should not use Auto-Numbers.  
Foundation Data or Source Data should avoid Auto-Numbers. If Auto-Numbers cannot be avoided, a Business Key must be identified.  
Employment related data should avoid Auto-Numbers unless a Business Key can be defined.
- Auto-Number ExternalCodes can be used without Business Key only in the following cases:
  - Where the objects are always inserted and never updated.  
Composite objects do not cause problems for instance sync because, the import uses Full Purge mode and composite objects are usually not referenced from other objects. However, updating and inserting composite records through incremental imports or through OData upsert can still be a challenge.

## 4.4 Adding Object Definitions to a Bundle for Transportation

Add an MDF object definition to a bundle so that you can transport the definition to a paired system using Configuration Transport Center.

### Prerequisites

You have created a transport bundle for the corresponding MDF object in Configuration Transport Center.

If the MDF object is secured, you have the necessary view and edit permission.

## Context

Bundles are artifacts in Configuration Transport Center that contain configurations of your system. You can use bundles to transport the configuration of a source system to a paired target system so that you don't need to manually configure it.

## Procedure

1. Go to [Configure Object Definitions](#) and open the definition of an MDF object.
2. Choose [Take Action](#) [Add to Transport Bundle](#) to add the current object definition to an existing bundle.  
A list of available transport bundles appears.
3. Select the bundle you want to add the configuration to and choose [Save](#).

## Results

The object definition is added to the transport bundle.

## Next Steps

You can now view, download, or transport the configuration bundle in Configuration Transport Center.

## Related Information

[Exporting SAP SuccessFactors Application Configurations Using Configuration Transport Center](#)  
[Adding Configurations to the Bundle using Configuration Transport Center](#)

## 4.5 Additional Object Configurations

This section outlines the additional configurations that can be applied to individual MDF objects to modify their behavior.

You can implement these changes in the [Manage Data](#) [Object Configuration](#) section under [Admin Center](#). From the second drop down menu, you can select the object you want to modify.

If the desired object is not listed in the second drop down, use the [Create New](#) option to create a new record for [Object Configuration](#) and choose the relevant object type.

You can modify the following attributes in the *Object Configuration* section:

Additional Attributes for MDF Object Configuration

| Attribute                              | Purpose   | Default Value  | Recommended Value | Additional Details  |
|--|---|----------------|-------------------|---|
| <i>Object Type</i>                     | Object for which all the other attributes are applied.  | Not Applicable | Not Applicable    | The drop down shows only root level object (All objects with effective dating None, Basic, Multiple Changes Per Day)  |
| <i>Validate Field Criteria</i>         | The option enables the validation of field criteria defined for fields with the data type Generic Object during <i>Save</i> operations performed using the Import or OData APIs. Field criteria on the UI are always validated, irrespective of this attribute. | No             | No                | It is recommended to keep the value as <i>No</i> and instead validate the field values offline before you proceed with the import.<br><br>Setting this value to <i>Yes</i> may lead to significant performance degradation during the <i>Import</i> or OData operations. The extent of the performance impact depends on the number of fields and criteria added. |
| <i>Disable Child Object In History</i> | The option enables you to hide child associations (both composite and Valid When) from the history panel on the <i>Manage Data</i> page or similar pages for a specific object.   | No             | No                | This is useful when there are numerous records at the composite child level.<br><br>When set to <i>Yes</i> , it can enhance the performance of loading the UI in the manage data section. However, you can't view the changes in the history panel, although the detailed view always display the updates.  |

| Attribute   | Purpose  | Default Value | Recommended Value | Additional Details   |
|---|--|---------------|-------------------|--|
| <a href="#">Disable Searchable Fields Config In Auto Complete</a> | The option allows to exclude all the search-able fields from the UI for search.  | No            | No                | <p>If the search-able fields are not required, they must be removed from the object definition entirely. Setting this to <a href="#">Yes</a> can significantly enhance search performance, as the search will be limited to external code and external name only.</p> <p>For more details, see <a href="#">Configuring the Disable Searchable Fields Config in Auto Complete [page 68]</a></p> |
| <a href="#">Enable Prefix-based Autocomplete Search</a>           | <p>By default, the search functionality in the <a href="#">Manage Data</a> page uses <a href="#">Contains</a> search.</p> <p>If this option is enabled, it switches to a <a href="#">Starts With</a> search.</p> | No            | No                | <p>The selection of this value depends on the business use case.</p> <p>Selecting <a href="#">Yes</a> displays fewer results but makes the search faster.</p>  |
| <a href="#">Rule Execution Criteria</a>                           | The option enables to decide whether the rules must be applied to all records or only to the ones that have been changed.  | No selection  | No selection      | For more details, see <a href="#">Assigning Rule Execution Criteria for an MDF Object [page 162]</a>   |

| Attribute   | Purpose   | Default Value  | Recommended Value | Additional Details   |
|---|---|----------------|-------------------|--|
| <a href="#">Enable Snapshot Pagination (OData V2)</a> | The option enables snapshot pagination at the object level.                               | No             | No                | <p>You can select this option to enable snapshot based pagination for this object. Note that this setting does not apply to its child objects. With this configuration, regular users (who lack the <a href="#">Admin access to MDF OData API</a> permission) can query the corresponding V2 API using snapshot pagination to prevent data duplication or missing records in the response.</p> <p>To enable snapshot-based pagination for all MDF objects and their corresponding child objects, the <a href="#">Admin access to MDF OData API</a> permission is required.</p> |
| <a href="#">Disable Check for Non-editable Fields</a> | The option is applicable only to the Latest People Profile to enable stricter changes.    | No             | No                | <p>It is recommended not to change this attribute value.</p> <p>If you enable this attribute, a strict check of the RBP configuration for the specific MDF object is enforced.</p>   |
| <a href="#">Manage Data On Load Rules</a>             | The option allows to add one or more rules that are executed when the records are loaded. | Not Applicable | Not Applicable    | <p>All the rules attached here are executed in the order they are defined within this field.</p> <p>However, if the <a href="#">onLoad</a> rules are also attached to the <a href="#">Configuration UI</a>, they take precedence, and the rules defined here aren't executed.</p>  |

## 4.6 Legislatively Sensitive Personal Data Configuration

It's required to have a flexible approach to enable different behavior and configurations based on specific legislation. MDF has defined a Legislative Sensitive Personal Data configuration called LSPDConfig, which indicates that an object contains personal data with respect to legislation.

LSPD configuration can be done using the [Configure Object Definition](#) page.

You can define this configuration for a parent object and open this configuration to edit the legislation-specific child object. If you're defining LSPD configuration for custom object types, Data Subject field must be a user field.

To configure LSPD, you must enter the *Module Name* (for example, Employee Central) and corresponding *Functional Area* (for example, Personal Information). Configuring the context field is optional. Context field can be a field from the Object Type or field from the hierarchy of that Object Type. If a field is part of a one-to-many ValidWhen association or is a multi-level ValidWhen reference field, then you cannot configure it as a context field.

### Note

Release 1H 2021 onwards, you can't modify Module Name and Functional Area field values for most of the predelivered objects. In earlier releases, if modifications were made to these field values in any predelivered object, it will be reset to the default value decided by the modules.

For more information about the supported combination of Modules and Functional Area in MDF Custom Objects, refer to *Modules That Support DRTM Purge of MDF Custom Objects in Setting Up and Using Data Protection and Privacy* guide.

The screenshot shows the 'Configure Object Definitions' interface for 'Legislatively Sensitive Data Configuration'. At the top, there is a search bar with 'Legislatively Sensitive Data C...' and a dropdown menu. Below the search bar, there are several configuration sections:

- Object Type:** A dropdown menu with 'No Selection'.
- Data Subject Field:** A dropdown menu with 'No Selection'.
- Data Subject Field Type:** A dropdown menu with 'No Selection'.
- Status:** A dropdown menu with 'Active'.
- Data Protection and Privacy Configuration:** A section with a trash icon and a plus icon. It contains:
  - Module Name:** A dropdown menu with 'No Selection'.
  - Functional Area:** A dropdown menu with 'No Selection'.
- Context Fields:** A section with a trash icon and a plus icon. It contains:
  - Field Name:** A dropdown menu with 'No Selection'.
  - Status:** A dropdown menu with 'Active'.
  - contextFieldLabel:** A text input field with the value 'Click or focus to edit'.
- Read Access Log Configuration:** A section with a trash icon and a plus icon. It contains:
  - Field Name:** A dropdown menu with 'No Selection'.
  - Enable Logging:** A dropdown menu with 'No'.
  - Status:** A dropdown menu with 'Active'.
- Russian Data Privacy Configuration:** A section with a trash icon and a plus icon. It contains:
  - Is Russian Data Privacy Enabled:** A dropdown menu with 'No'.

At the bottom right, there are 'Cancel' and 'Save' buttons.

For additional information about LSPD configuration, refer to *Defining a Custom Object as Containing Sensitive Data for a Given Module*.

## Related Information

[Modules That Support DRTM Purge of MDF Custom Objects](#)

[Defining a Custom Object as Containing Sensitive Personal Data for a Given Module](#)

## 4.7 Configuring Fields for Read Access Logging

You can include a field for read audit by enabling read access logging in the legislative sensitive data configuration of the the object.

### Prerequisites

The legislative sensitive data configuration for the object exists and is in active status. If it doesn't exist, create one.

### Context

You can choose to configure and enable read access logging for custom object fields. However, in predelivered objects, only a few fields are by default configured as reading access logging fields. You can enable read access logging for them using the Legislatively Sensitive Data Configuration page.

You can't enable read access logging for the following scenarios:

- If the field is of data type Auto Number, Auto Complete, or Data Source.
- If the field is a system field.
- If the field is configured as a searchable field, context field, or a value help parameter.
- If the field is one of the standard fields: `externalCode`, `externalName`, `effectiveStartDate`, `effectiveEndDate`, `transactionSequence`, and `effectiveLatestChange`.

### Procedure

1. Go to [Admin Center > Configure Object Definitions](#).
2. Select [Legislative Sensitive Data Configuration](#) from the [Search](#) dropdown. Search for the object for which you want to configure the read audit field. The Legislatively Sensitive Data Configuration page is displayed.



3. Choose *Take Action > Make Correction*.
4. Go to *Read Access Log Configuration* section and choose the field for which you want to configure read access logging from the *Field Name* dropdown. You can also enable logging for this field by choosing *Yes* from the *Enable Logging* dropdown.

For custom objects, all the composite association fields are listed in the *Field Name* dropdown.

5. Save your changes.

## Results

You've enabled the field for read access logging. Access to this field is now included read audit reports.

## 4.8 Custom Field Limits on Pre-delivered and Custom MDF Generic Objects

There are limits on how many custom fields can be added for pre-delivered standard generic objects and custom generic objects. For more information about the default limits, see [Important Configuration Guidelines and Usage Limits \[page 10\]](#).

### MDF Data Type Mapping

The following table lists the mapped MDF data types:

MDF Data Type and its Mapping

| MDF Data Type | Mapped To... |
|---------------|--------------|
| String        | String       |
| Number        | Number       |
| Auto Number   | Number       |
| Decimal       | Decimal      |
| Boolean       | Number       |
| Date          | Date         |
| DateTime      | Date         |
| Time          | Date         |

| MDF Data Type     | Mapped To... |
|-------------------|--------------|
| Picklist          | Number       |
| Translatable      | Number       |
| Datasource        | String       |
| Enum              | String       |
| Generic Object    | Number       |
| Foundation Object | Number       |
| Attachment        | Number       |
| User              | String       |
| CLOB              | Number       |

### Caution

The values for fields of data type Auto Number are populated by the system. We don't recommend editing these fields manually because it can cause data corruption and disturb the auto number sequence populated by the system.

# 5 Managing MDF Object Instance

Using the Manage Data page, you can create instances of your MDF object. This instance allows you to populate the MDF object with data.

As an Admin, you can modify data for an existing object instance or you can create an instance for your object.

## [Creating an MDF Object Instance \[page 64\]](#)

After creating an MDF Object, you can create instances for this object.

## [Viewing MDF Object Data \[page 65\]](#)

Using Manage Data page, you can view the data created for the object instances.

## [Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)

You can copy the configuration and non-user transactional data of an MDF record to create a new record in *Manage Data* so that you don't have to enter all fields manually.

## [Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)

## [Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)

## [Forward Propagation \[page 74\]](#)

This section explains forward propagation in the MDF.

## [Improved Attachment Handling for MDF \[page 77\]](#)

Based on compliance requirements, the attachments for MDF entities belonging to users in Russian legal entity and relevant for Russian Data Privacy Legislation are now saved first to Russian Data Center and then to Global Data Centers.

## [Viewing MDF Object Data Volumes \[page 78\]](#)

Understand the data consumptions of the MDF objects in your instance to help you make decisions on whether or when to purge MDF data.

## [Adding Object Instances to a Transport Bundle \[page 80\]](#)

Add an MDF object instance to a transport bundle so that you can transport the instance to a paired system using Configuration Transport Center.

## 5.1 Creating an MDF Object Instance

After creating an MDF Object, you can create instances for this object.

### Prerequisites

You have created an object definition.

You have read and understand the guidelines and limits described in [Important Configuration Guidelines and Usage Limits \[page 10\]](#).

## Context

As an Admin, you can create multiple instances for an MDF object.

## Procedure

1. Go to [Admin Center > Manage Data](#).
2. From the *Create New* dropdown, select the object definition you want to create an instance for.

The *Create New* dropdown displays the object definitions that are available on your instance. If you do not see an object definition that you expected to see, check the security permissions. For more information about security, see the [Configuring the Object Definition > Adding Security section](#).

Fields corresponding to the selected object definition appears on the UI.

3. Provide the values for all the mandatory fields and click *Save*.

**Task overview:** [Managing MDF Object Instance \[page 64\]](#)

## Related Information

[Viewing MDF Object Data \[page 65\]](#)

[Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)

[Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)

[Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)

[Forward Propagation \[page 74\]](#)

[Improved Attachment Handling for MDF \[page 77\]](#)

[Viewing MDF Object Data Volumes \[page 78\]](#)

[Adding Object Instances to a Transport Bundle \[page 80\]](#)

## 5.2 Viewing MDF Object Data

Using Manage Data page, you can view the data created for the object instances.

## Context

As an Admin, you can search for the data created corresponding to any object using the **Search** option.

## Note

- We have a limitation on the number of one-to-many composite association child records that can be displayed on the Manage Data page. We're limiting it to 3000 records. If there are more than 500 child records for an object, we recommend that you use the Import and Export Data option to view or modify the records.
- When you search for fields such as External Code or External Name, you can view all records containing even a part of the search text. The order of the words in search text doesn't change the search result. For example, if you searched for **Sales Marketing**, and there are two searchable fields External Code and External Name. Then, the search result displays both records where External Code contains Sales and Marketing and records where External Name contains Sales and Marketing.
- For the effective-dated MDF objects, the search results are always displayed based on the timezone of the user that is set in the Employee Profile.
- The timestamp on the Manage Data page now displays the user ID or assignment ID of the individual who last modified the object instance. Previously, when you wanted to change user IDs in cases such as company acquisition or employee relocation, a support ticket was needed. The user ID conversion process was costly and time-consuming. Now, you can use assignment ID to define the relationship between a person and a company.
- If an attachment is displayed with the **Marked for Deletion** prefix text, we recommend that you download and upload the attachment once more, to prevent data loss. Report this issue to Technical Support for further investigation.

## Procedure

1. Go to [Admin Center](#) > [Manage Data](#).
2. From the *Search* dropdown, select the object definition for which you want to view the data for.  
The second dropdown gets populated with instances created for the selected object.
3. Select any object instance from the second dropdown.  
UI displays all the data created corresponding to the selected object instance.
4. To include inactive records, choose **Yes** from the *Include Inactives* dropdown.  
UI displays the *Search As Of* dropdown, if you choose objects with effective dating properties.
5. To modify the existing data, go to [Take Action](#) > [Make Correction](#) and once done, click *Save*.

**Task overview:** [Managing MDF Object Instance \[page 64\]](#)

## Related Information

[Creating an MDF Object Instance \[page 64\]](#)

[Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)

[Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)

[Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)

[Forward Propagation \[page 74\]](#)

[Improved Attachment Handling for MDF \[page 77\]](#)

[Viewing MDF Object Data Volumes \[page 78\]](#)

[Adding Object Instances to a Transport Bundle \[page 80\]](#)

## 5.3 Copying Configuration and Non-User Transactional MDF Data

You can copy the configuration and non-user transactional data of an MDF record to create a new record in *Manage Data* so that you don't have to enter all fields manually.

### Prerequisites

You have the [Administrator](#) > [Metadata Framework](#) > [Manage Data](#) permission.

To copy the data of a secured MDF object, you also need the View and Create permission if the object is effective-dated, and View and Edit permissions if it's non-effective dated.

### Context

Data copy can be useful when you need to create a new record that share the same field values with an existing one. Please note that the copy function is only available for MDF objects that are categorized as *Configuration* or *Non-User Transactional* objects.

#### Note

You can't change the *Object Category* field value of a predelivered MDF object.

Not all field values can be copied. Here's a list of the ones you can copy:

- Field values except external codes, business keys, autonumber fields, system fields, and attachments
- Associations, including both Valid When and Composite associations. For Valid When association, the copied record references to the same target record. For Composite association, a new record is created with the same field values.
- Order of composite childs are retained in the copied record

## Procedure

1. Go to [Manage Data](#), search for your MDF object, and open the record you want to copy from.
2. Choose [Take Action](#) [Copy](#). Confirm the dialog when popped up. The edit screen of the new record displays.

### → Tip

To copy a timeslice of an effective-dated record, choose the record first, and then choose [Take Action](#) [Copy](#). If you have multiple timeslices to copy, copy them one by one.

3. Enter the required fields.

Fields that can't be copied, such as external code, business keys, and attachments, are empty in the new record.

Values for system field and autonumber fields are handled the same way you create a new record.

4. Save your changes.

**Task overview:** [Managing MDF Object Instance \[page 64\]](#)

## Related Information

[Creating an MDF Object Instance \[page 64\]](#)

[Viewing MDF Object Data \[page 65\]](#)

[Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)

[Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)

[Forward Propagation \[page 74\]](#)

[Improved Attachment Handling for MDF \[page 77\]](#)

[Viewing MDF Object Data Volumes \[page 78\]](#)

[Adding Object Instances to a Transport Bundle \[page 80\]](#)

## 5.4 Configuring the Disable Searchable Fields Config in Auto Complete

### Prerequisites

You have the [Administrator Permissions](#) [Metadata Framework](#) [Manage Data](#) permission.

## Context

A new field *Disable Searchable Fields Config In Auto Complete* is added to the existing *Object Configuration* object in the *Manage Data* page, which is used to store additional metadata for object definition. By default, the field value is set to **No**. This reflects the default behavior of MDF auto complete, which picks up all the searchable fields configuration from object definition for auto complete search. The large number of searchable fields result in performance degradation.

If there's no instance for *Object Configuration*, you can create an instance for the required object.

### Note

The *Object Configuration* instance must be created only for a required MDF object. For example, **Position** object in high volume scenarios.

Once the Object Configuration instance is created, set *Disable Searchable Fields Config In Auto Complete* value to **Yes** to avoid picking up all the searchable field configuration for auto complete search. This uses minimal fields such as *externalCode* and *externalName* in the auto complete search and enhances the performance.

The following procedure is an example to show how to configure *Disable Searchable Fields Config In Auto Complete* for *Position* object to improve performance.

## Procedure

1. In the *Manage Data* admin tool, select *Object Configuration* option for *Create New*.
2. Select the *Object Type* as *Position*.
3. Select **Yes** for *Disable Searchable Fields Config In Auto Complete*.
4. Click **Save**.

### Note

Set *Disable Searchable Fields Config In Auto Complete* value to **No** to pick up all the searchable fields.

## Results

In the *Manage Data* page, access *Position* record from the search list box. The *Position* records are filtered based on the default fields *externalName* and *externalCode* and not on any other searchable fields defined in the object level, thus improving the performance.

**Task overview:** [Managing MDF Object Instance \[page 64\]](#)



## Related Information

[Creating an MDF Object Instance \[page 64\]](#)

[Viewing MDF Object Data \[page 65\]](#)

[Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)

[Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)

[Forward Propagation \[page 74\]](#)

[Improved Attachment Handling for MDF \[page 77\]](#)

[Viewing MDF Object Data Volumes \[page 78\]](#)

[Adding Object Instances to a Transport Bundle \[page 80\]](#)

## 5.5 Referring to an RBP Subject User Field on the Referred Node

### Context

Previously, RBP subject user field could only refer to a user field with in that entity. It was not possible to refer to a user field of referenced generic object. To refer to an RBP subject user field that is on a referred node, consider the following example and proceed as follows.

### Procedure

1. Add a user type field **cust\_user** (as required) to the **cust\_subGO2** object in the [Object Definition](#) UI.

## Object Definition: cust\_subGO2 (cust\_subGO2)

\* Code **cust\_subGO2**  
\* Effective Dating **Basic**  
API Visibility **Not Visible**  
\* Status **Active**  
MDF Version History **No Selection**  
defaultScreen  
Label **cust\_subGO2** ⓘ  
Description  
API Sub Version **V1.1**  
Subject User Field  
Workflow Routing  
Pending Data **No**  
Todo Category **Generic Object Change Requests**


### Fields

| Name               | Database Field Name | Maximum Length | Data Type | <a href="#">(21) More</a> |
|--------------------|---------------------|----------------|-----------|---------------------------|
| externalCode       | externalCode        | 128            | String    | <a href="#">Details</a>   |
| externalName       | externalName        | 128            | String    | <a href="#">Details</a>   |
| effectiveStartDate | effectiveStartDate  | 255            | Date      | <a href="#">Details</a>   |
| cust_user          | sfFields.sfField1   |                | User      | <a href="#">Details</a>   |

### Adding User Type Field

2. Add a generic object field **cust\_GO** (as required) to the **cust\_GO2** object in the *Object Definition* UI.

## Object Definition: cust\_GO2 (cust\_GO2)

\* Code **cust\_GO2**  
 \* Effective Dating **Basic**  
 API Visibility **Not Visible**  
 \* Status **Active**  
 MDF Version History **No Selection**  
 defaultScreen  
 Label **cust\_GO2**   
 Description  
 API Sub Version **V1.1**  
 Subject User Field  
 Workflow Routing  
 Pending Data **No**  
 Todo Category **Generic Object Change Requests**

**Fields**

| Name               | Database Field Name | Maximum Length | Data Type      | <a href="#">(21) More</a> |
|--------------------|---------------------|----------------|----------------|---------------------------|
| externalCode       | externalCode        | 128            | String         | <a href="#">Details</a>   |
| externalName       | externalName        | 128            | String         | <a href="#">Details</a>   |
| effectiveStartDate | effectiveStartDate  | 255            | Date           | <a href="#">Details</a>   |
| cust_GO            | sfFields.sfField1   |                | Generic Object | <a href="#">Details</a>   |

### Adding Generic Object Field

3. Create instances for **cust\_subGO2** and **cust\_GO2** in the *Manage Date* page.

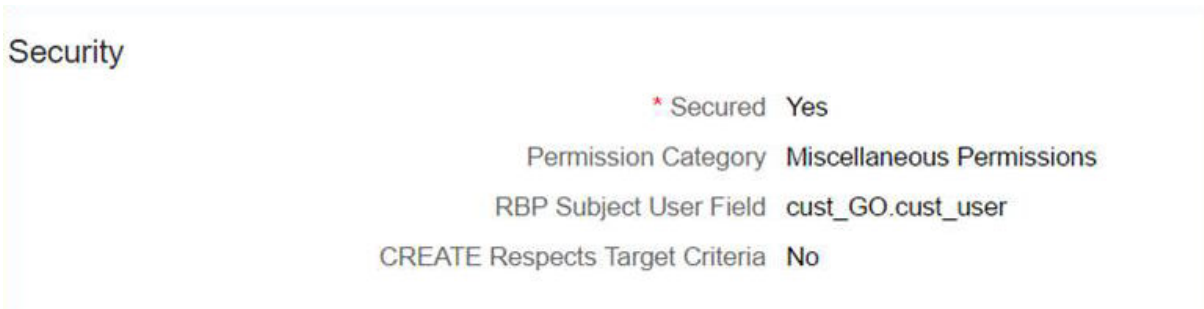
| externalCode | cust_user | effectiveStartDate of two records |
|--------------|-----------|-----------------------------------|
| Root1        | admin     | 01/01/2000; 01/01/2011            |
| Root2        | cgant1    | 01/01/2000; 01/01/2011            |
| Root3        | jlo       | 01/01/2000; 01/01/2011            |

Creating Instance for cust\_subGO2

| externalCode | cust_user | effectiveStartDate of two records |
|--------------|-----------|-----------------------------------|
| OBJ1         | Root1     | 01/01/2000; 01/01/2011            |
| OBJ2         | Root2     | 01/01/2000; 01/01/2011            |
| OBJ3         | Root3     | 01/01/2000; 01/01/2011            |

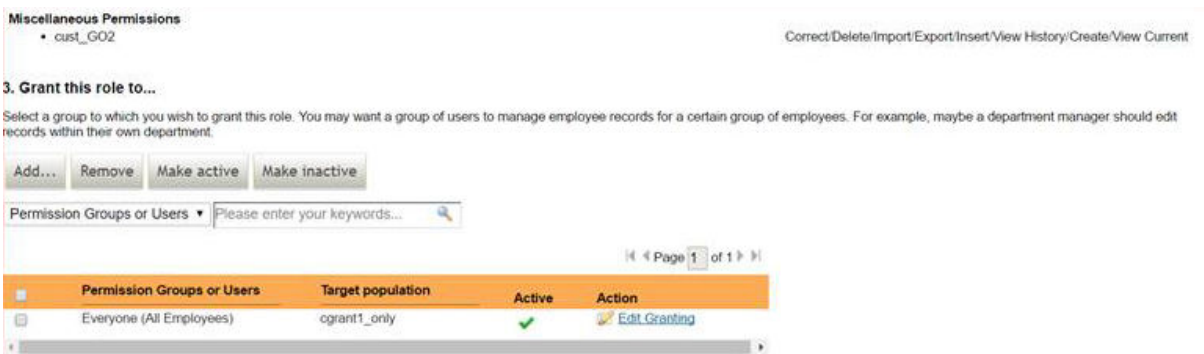
Creating Instance for cust\_GO2

4. Set the *RBP Subject User Field* of **cust\_GO2** to **cust\_GO.cust\_user**.



#### Setting RBP Subject User Field

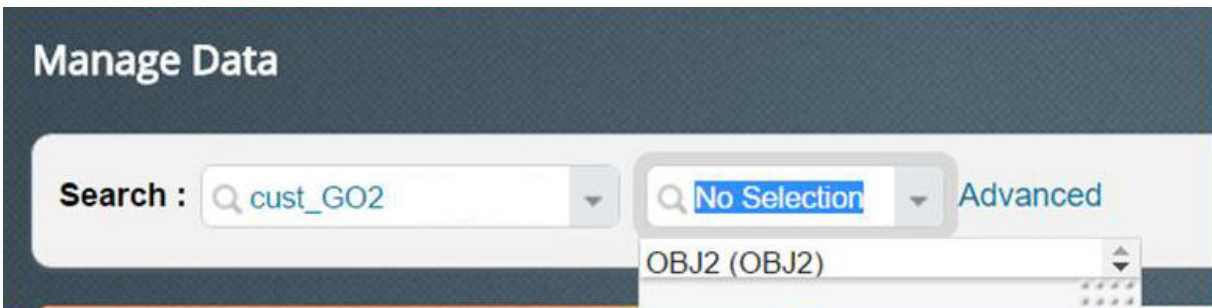
- Set the Role-based permission.



#### Set Role-based Permission for cust\_GO2

- Search for **cust\_GO2** in the *Manage Data* page to verify whether the GO referenced user field based RBP subject user field works correctly.

This displays, OBJ2.



#### Verifying GO Referenced User Field

#### Note

This feature also supports person field reference GO and OData.

**Task overview:** [Managing MDF Object Instance \[page 64\]](#)

## Related Information

[Creating an MDF Object Instance \[page 64\]](#)

[Viewing MDF Object Data \[page 65\]](#)

[Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)

[Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)

[Forward Propagation \[page 74\]](#)

[Improved Attachment Handling for MDF \[page 77\]](#)

[Viewing MDF Object Data Volumes \[page 78\]](#)

[Adding Object Instances to a Transport Bundle \[page 80\]](#)

## 5.6 Forward Propagation

This section explains forward propagation in the MDF.

Forward propagation of future records means that a change in the value of a field in an object is also made (“propagated”) to future records for the same object. The forward propagation of this field change stops as soon as one of the future records has a field value maintained that is different than the original field value.

Forward propagation is also triggered for effective updates (insertions), and in some cases for corrections or deletions. Each object can exhibit different behaviors for forward propagation. Based on the action performed, correction or deletion, the object may or may not support it.

Please refer to the module documentation to verify if forward propagation is supported and its behavior when correction or deletion as an action is performed.

### Note

Custom Objects do not support forward propagation.

MDF supports forward propagation for the following:

- Associations and composite children
- Imports
- Picklists
- Translatable Fields

Forward propagation treats a deletion that same way as a correction, for example, when a time slice is deleted, it will have the same values as the previous time slice. When a timeslice is deleted, it behaves the same as correcting the 'deleted' row to have exactly the same values as the previous timeslice.

## Example

In this example, the company, employee class and effective start date are changed for an employee. TestPosition is an MDF object that has implemented the forward propagation On Delete behavior.

The 1st time slice for the TestPosition is created on Oct 18 with the value of the company as 11 and employeeClass as L1. The next time slice for the TestPosition is created on Oct 21 with the value of the company as 12 and employeeClass as L2. You can see that the original values are struck through to denote the changes.

Back to: Admin Center

## Manage Data

**Successfully Saved**

Search :   [Advanced](#) Create New :

**History** «

|   |
|---|
| <b>10/21/2016</b> <a href="#">Take Action</a> |
| employeeClass: L2 (L2) L1-(L1)                |
| company: 12 (NewAgain) 11-(NewCompany)        |

|                              |
|------------------------------|
| <b>10/18/2016</b>            |
| TestPosition: Record created |

**TestPosition: LP1 (LP1)** [Insert New Record](#)

- \* code LP1
- externalName LP1
- \* effectiveStatus Active
- \* effectiveStartDate 10/21/2016 40/18/2016
- positionTitle
- criticality
- comment
- incumbent
- changeReason
- description
- jobTitle
- jobLevel T1 (T1)
- employeeClass L2 (L2) L1-(L1)
- regularTemporary
- targetFTE
- vacant No
- company 12 (NewAgain) 11-(NewCompany)

Next we insert time slice for the TestPosition on Oct 19 with the value of the company as 13 and employeeClass as L1.

Back to: Admin Center

## Manage Data

**Successfully Saved**

Search :   [Advanced](#) Create New :

**History** «

|  |
|--|
| <b>10/21/2016</b>                      |
| employeeClass: L2 (L2) L1-(L1)         |
| company: 12 (NewAgain) Company-13-(13) |

|   |
|---|
| <b>10/19/2016</b> <a href="#">Take Action</a> |
| company: Company 13 (13) 11-(NewCompa...      |

|                              |
|------------------------------|
| <b>10/18/2016</b>            |
| TestPosition: Record created |

**TestPosition: LP1 (LP1)** [Insert New Record](#)

- \* code LP1
- externalName LP1
- \* effectiveStatus Active
- \* effectiveStartDate 10/19/2016 40/18/2016
- positionTitle
- criticality
- comment
- incumbent
- changeReason
- description
- jobTitle
- jobLevel T1 (T1)
- employeeClass L1 (L1)
- regularTemporary
- targetFTE
- vacant No
- company Company 13 (13) 11-(NewCompany)

Now for the time slice for TestPosition on Oct 21, -the struck through values are shown as that of the Oct 19 time slice with the company as 13 and employeeClass as L1.

Back to: Admin Center

### Manage Data

Search :    Create New :

**History «**

- 10/21/2016  
employeeClass: L2 (L2) L+ (L+)  
company: 12 (NewAgain) Company-13 (13)
- 10/19/2016  
company: Company 13 (13) 11 (11)
- 10/18/2016  
TestPosition: Record created

**TestPosition: LP1 (LP1)** Insert New Record

- \* code LP1
- externalName LP1
- \* effectiveStatus Active
- \* effectiveStartDate 10/19/2016 40/48/2016
- positionTitle
- criticality
- comment
- incumbent
- changeReason
- description
- jobTitle
- jobLevel T1 (T1)
- employeeClass L1 (L1)
- regularTemporary
- targetFTE
- vacant No
- company Company 13 (13) 11 (11) 11 (NewCompany)

Now we delete the time slice for the TestPosition on Oct 19. Since the time slice for Oct 19 is deleted and no longer there, forward propagation on delete comes into play. For the time slice for TestPosition on Oct 21, the struck through values are shown as that of the time slice for Oct 18 with the company as 11 and employeeClass as L1.

Back to: Admin Center

### Manage Data

Successfully Deleted

Search :    Create New :

**History «**

- 10/21/2016  
employeeClass: L2 (L2) L+ (L+)  
company: 12 (NewAgain) 11 (NewCompany)
- 10/18/2016  
TestPosition: Record created

**TestPosition: LP1 (LP1)** Insert New Record

- \* code LP1
- externalName LP1
- \* effectiveStatus Active
- \* effectiveStartDate 10/21/2016 40/48/2016
- positionTitle
- criticality
- comment
- incumbent
- changeReason
- description
- jobTitle
- jobLevel T1 (T1)
- employeeClass L2 (L2) L+ (L+)
- regularTemporary
- targetFTE
- vacant No
- company 12 (NewAgain) 11 (NewCompany)

Parent topic: [Managing MDF Object Instance \[page 64\]](#)

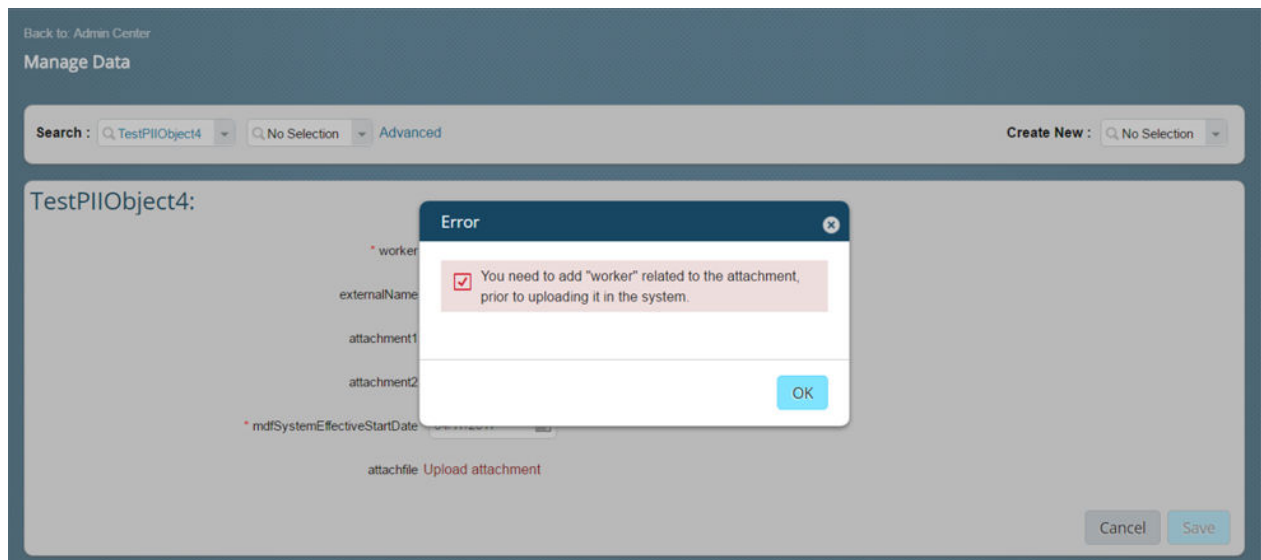
## Related Information

- [Creating an MDF Object Instance \[page 64\]](#)
- [Viewing MDF Object Data \[page 65\]](#)
- [Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)
- [Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)
- [Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)
- [Improved Attachment Handling for MDF \[page 77\]](#)
- [Viewing MDF Object Data Volumes \[page 78\]](#)
- [Adding Object Instances to a Transport Bundle \[page 80\]](#)

## 5.7 Improved Attachment Handling for MDF

Based on compliance requirements, the attachments for MDF entities belonging to users in Russian legal entity and relevant for Russian Data Privacy Legislation are now saved first to Russian Data Center and then to Global Data Centers.

The importance of this improvement is with regards to the privacy legislation compliance and this feature is enabled for delivered MDF objects where an object level flag is turned on to indicate that the object can contain legislatively sensitive personal data, for example, Background Information object from Employee Central. This feature ensures that for the concerned object, the subject user field has to be provided prior to uploading an attachment.



Parent topic: [Managing MDF Object Instance \[page 64\]](#)



## Related Information

[Creating an MDF Object Instance \[page 64\]](#)

[Viewing MDF Object Data \[page 65\]](#)

[Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)

[Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)

[Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)

[Forward Propagation \[page 74\]](#)

[Viewing MDF Object Data Volumes \[page 78\]](#)

[Adding Object Instances to a Transport Bundle \[page 80\]](#)

## 5.8 Viewing MDF Object Data Volumes

Understand the data consumptions of the MDF objects in your instance to help you make decisions on whether or when to purge MDF data.

### Prerequisites

To check the data volumes of an MDF object, you need the following permissions:

- [Administrator Permissions](#) > [Metadata Framework](#) > [Manage Data](#) >
- View permission of the following MDF objects under [User Permissions](#) > [Miscellaneous Permissions](#) >:
  - MDF Audit Data Volume
  - MDF Association Data Volume
  - MDF Transactional Data Volume

To view the exported data volume information, you also need the [Administrator Permissions](#) > [Admin Center Permissions](#) > [Monitor Scheduled Jobs](#) > permission.

### Context

You can use the following new MDF objects and specify another MDF object in [Manage Data](#) to view its data volume:

- MDF Audit Data Volume: shows the volume of audit data available for a given MDF object.
- MDF Association Data Volume: shows the volume of association data available for a given MDF object.
- MDF Transactional Data Volume: shows the volume of transactional data available for a given MDF object. Transactional data refers to the day-to-day business data that's available in a system.

When viewing MDF data volumes, keep the following in mind:

- The data volume objects and the corresponding instances are read-only.
- Data volumes are collected monthly for every company through the MDF Data Volume Collection job and is available for viewing with proper permission. With every job run, a new record with the latest data volume is created.
- There's no restriction on what types of objects you can check the data volume for as long as you have the right permission, including technical objects and system objects. However, because you don't have control over the data of such objects, you can ignore them.

## Procedure

1. Go to *Admin Center* and open *Manage Data* from Tool Search.
2. In *Manage Data*, choose one of the following objects from Search:
  - MDF Audit Data Volume
  - MDF Transactional Data Volume
  - MDF Association Data Volume
3. Select an MDF object instance to view its corresponding data volume. Choose the question icon (❓) to understand what each field means.
4. Alternatively, you can also go to *Import and Export Data*, select an MDF data volume object, and export the data volume information. This option allows you to aggregate data volumes of multiple objects and analyze data trends using a spreadsheet application offline.

## Next Steps

The volume of MDF data can impact how your system perform. We recommend that you regularly check the data consumption of your instance against the MDF configuration guidelines and usage limits, and purge the data that you no longer need. For more information about usage limits and purging data, see the Related Information section.

**Task overview:** [Managing MDF Object Instance \[page 64\]](#)

## Related Information

[Creating an MDF Object Instance \[page 64\]](#)

[Viewing MDF Object Data \[page 65\]](#)

[Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)

[Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)

[Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)

[Forward Propagation \[page 74\]](#)

[Improved Attachment Handling for MDF \[page 77\]](#)

[Adding Object Instances to a Transport Bundle \[page 80\]](#)

[Purging MDF Data \(Non-DRTM\) \[page 206\]](#)

[Important Configuration Guidelines and Usage Limits \[page 10\]](#)

## 5.9 Adding Object Instances to a Transport Bundle

Add an MDF object instance to a transport bundle so that you can transport the instance to a paired system using Configuration Transport Center.

### Prerequisites

- You've created a transport bundle for the corresponding MDF object in Configuration Transport Center.
- If the MDF object is secured, you have the necessary view and edit permission.

### Context

Bundles are artifacts in Configuration Transport Center that contain configurations of your system. You can use bundles to transport the configuration of a source system to a paired target system so that you don't need to manually configure it.

### Procedure

1. Go to [Manage Data](#) page and open an MDF object instance.
2. Choose **Take Action** > [Add to Transport Bundle](#) to add the current object instance to an existing bundle.  
A list of available transport bundles are displayed.
3. Select the bundle you want to add the configuration to and choose [Save](#).  
Your configuration is successfully added to the transport bundle. A success message is displayed.
4. Choose [Close](#).

### Results

The object instance is added to the transport bundle.

**Task overview:** [Managing MDF Object Instance \[page 64\]](#)

## Related Information

[Creating an MDF Object Instance \[page 64\]](#)

[Viewing MDF Object Data \[page 65\]](#)

[Copying Configuration and Non-User Transactional MDF Data \[page 67\]](#)

[Configuring the Disable Searchable Fields Config in Auto Complete \[page 68\]](#)

[Referring to an RBP Subject User Field on the Referred Node \[page 70\]](#)

[Forward Propagation \[page 74\]](#)

[Improved Attachment Handling for MDF \[page 77\]](#)

[Viewing MDF Object Data Volumes \[page 78\]](#)

[Exporting SAP SuccessFactors Application Configurations Using Configuration Transport Center](#)

[Adding Configurations to the Bundle using Configuration Transport Center](#)

# 6 Manage Configuration UI

You can use the *Manage Configuration UI* tool to create and modify UI blocks that appear on the *Manage Data* page, in *People Profile*, and *Workflow Details*.

As an Admin, you can use the *Manage Configuration UI* tool for tasks like:

- Change label for fields.
- Add new fields.
- Change the display sequence of fields.
- Change the layout of UI.
- Change visibility of fields.
- Set fields to required.
- Add Rules.
- Delete a field from UI.
- Add a link.
- Add a group.
- Add a reference. You can use it to add One to Many Valid When association of object definition.

## Note

Even though some modifications, such as changing field labels, can be done in both the Configuration UI and the *Configure Object Definition* page, it's recommended to do these updates primarily in the *Configure Object Definition* page. Use the Configuration UI for such modifications only when the functionality is not available in the *Configure Object Definition* page.

1. [Creating a Configurable UI \[page 83\]](#)  
A configurable UI allows you to define the fields and their layout as required by your company.
2. [Adding Configurable UI to a Bundle for Transportation \[page 84\]](#)  
Add a Configurable UI to a bundle so that you can transport the Configuration UI to a paired system using Configuration Transport Center
3. [Linking User Interface to Object Definition \[page 85\]](#)  
After creating the new UI, link it to an object definition. Once linked, the new UI appears on the *Manage Data* page and the *Workflow Details* page.
4. [Known Limitations and Considerations \[page 85\]](#)  
Describes the limitations and things to consider while using Configuration UI.

## Related Information

[Changing Field Attributes on the User Interface \[page 181\]](#)

## 6.1 Creating a Configurable UI

A configurable UI allows you to define the fields and their layout as required by your company.

### Context

Configurable UI can be created for any of the object in your instance. Also, multiple configurable UIs can be created for same base object.

### Procedure

1. Go to [Admin Center](#) > [Manage Configuration UI](#).
2. In the the *Id* field, provide a unique id that identifies the configurable UI.
3. From the *Select Base Object* dropdown, select an object type.

Based on the selected *Base Object*, the fields on the UI gets updated.

4. Now, you can configure the properties such as **Layout, Field Visibility, Adding Fields, Deleting Fields, Adding Rules**
5. Click [Save](#).

### Next Steps

After creating a configurable UI, next step is to link this UI using [Admin Center](#) > [Configure Object Definition](#) page.

**Task overview:** [Manage Configuration UI \[page 82\]](#)

**Next task:** [Adding Configurable UI to a Bundle for Transportation \[page 84\]](#)

## 6.2 Adding Configurable UI to a Bundle for Transportation

Add a Configurable UI to a bundle so that you can transport the Configuration UI to a paired system using Configuration Transport Center

### Prerequisites

- You've created a transport bundle in Configuration Transport Center.
- To use a configurable UI, you have the [▶ Administrator Permissions](#) [▶ Metadata Framework](#) [▶ Manage Configuration UI](#) [▶](#) permission.
- You have the [▶ Administrator Permissions](#) [▶ Configuration Transport Center](#) [▶ Access to transport configurations](#) [▶](#) permission.

### Context

Bundles are artifacts in the Configuration Transport Center that contain configurations of your system. You can use bundles to transport the configuration of a source system to a paired target system so that you don't need to manually configure it. You create a Configurable UI for any of the object in your instance and add it to a transport bundle.

### Procedure

1. Go to [▶ Admin Center](#) [▶ Manage Configuration UI](#) [▶](#).
2. In the *Id* field, search for a configurable UI with a unique id and *Select Base Object* field. Based on the selected *Base Object*, the fields on the UI get updated.
3. The *Add to Transport Bundle* option is available at the top right of the page in the *Manage Configuration UI* page. Choose *Add to Transport Bundle* option to add the selected Configurable UI to an existing bundle.

A list of available transport bundles appears.

4. Select the bundle that you want to add the configuration to and choose *Save*.

**Task overview:** [Manage Configuration UI \[page 82\]](#)

**Previous task:** [Creating a Configurable UI \[page 83\]](#)

**Next task:** [Linking User Interface to Object Definition \[page 85\]](#)

## 6.3 Linking User Interface to Object Definition

After creating the new UI, link it to an object definition. Once linked, the new UI appears on the [Manage Data](#) page and the [Workflow Details](#) page.

### Context

As an Admin, you can link the UI with Object Definition using [Admin Center](#) > [Configure Object Definitions](#) > page.

### Procedure

1. Go to [Admin Center](#) > [Configure Object Definitions](#) >.
2. From the [Search](#) dropdown, select [Object Definition](#) for which you've created the configuration.  
The second dropdown gets populated with values corresponding to the selected object.
3. Select an object definition from second dropdown.  
UI displays all the configurations corresponding to the selected object definition.
4. Click [Take Action](#) > [Make Correction](#) >.
5. From the [Default Screen](#) dropdown, select the configuration that you would like to attach.
6. Save your changes.

**Task overview:** [Manage Configuration UI \[page 82\]](#)

**Previous task:** [Adding Configurable UI to a Bundle for Transportation \[page 84\]](#)

**Next:** [Known Limitations and Considerations \[page 85\]](#)

## 6.4 Known Limitations and Considerations

Describes the limitations and things to consider while using Configuration UI.

- When using UI rules to update the state of nested groups in a one-to-many association, the rule applies only to the first row of the nested group.  
For example, if the condition in the rule references a field on the parent record (`parent.ParentField1 == 'A'`), the resulting action (setting `child.ChildField3.hidden=true`) is applied only to the first child record in the nested group. However, if the condition in the rule is based on a child record field (if



`child.ChildField1 == 'A' then set child.ChildField3.hidden = true`), then the rule is applied for all child records in the child record in the nested group.

- Any Fields marked as `visible = No` either through the field level Configuration on UI or using UI rules, may not work correctly if the fields are not displayed on the UI initially and user has to select the [Show more](#) option. This occurs when the [Visible Fields Count](#) is lower than the index of the hidden field. For example, if a child object has 10 fields, and the 7th field is marked as `visible = No`, but the Visible Fields Count is set to 5, the hidden field's behavior may be incorrect. To resolve this, increase the "Visible Fields Count" to 7 or higher.
- When working with child objects, avoid creating user defined groups. If such groups are created and UI rules are used to control the field states (such as visibility or required status), these rules will only work correctly for the first row of child objects. Instead, always use the [Collection Resolver](#) to define composite child relationships in Configuration UI.
- Some of the properties present in Configuration UI doesn't work on People Profile.
- At times, the reordering and other changes that are done on Configuration UI don't take effect on People Profile.
- When you delete an MDF record that has an associated workflow, an approval based on the workflow is required. A message on UI stating **This record has been marked for deletion. Deletion is pending approval** is displayed corresponding to the deleted object in Manage Configuration UI page.
- While adding a Reference Group, if the Reference Resolver value isn't specified then the Reference Group isn't saved.
- The **RTE editor** has been updated and now has different behavior.
- You must not use the [Source](#) option in the [RTE editor](#)
- Formatting text in the [RTE editor](#) adds hidden characters (HTML/markup), that contribute to the character limit defined for that field. This code isn't visible, but it does count towards the maximum character limit defined for that field. Therefore, you must increase the limit accordingly (three to four times the visible text length) to accommodate formatting. For more details, see [Rich Text Editor](#) in the Related Information section.

**Parent topic:** [Manage Configuration UI \[page 82\]](#)

**Previous task:** [Linking User Interface to Object Definition \[page 85\]](#)

## Related Information

[Rich Text Editor](#) 

# 7 MDF Configurations for Latest People Profile

The latest People Profile now comes with Full Profile, a revamped data hub that consolidates all kinds of data for an employee in one single place.

This table shows the required MDF configurations and updates for the latest People Profile.

## MDF-Related Changes to Latest People Profile

| MDF Changes in Latest People Profile | Description   | Notes  |
|--------------------------------------|---|--|
| Custom MDF Objects                   | <ul style="list-style-type: none"> <li>Supports only secured custom MDF Objects that are marked as <b>Yes</b> for the Secured field on the object definition.</li> <li>No support for any custom MDF object which has more than two levels of composite (Parent plus two level of composites are supported).</li> </ul> | <ul style="list-style-type: none"> <li>For pre-delivered MDF objects, there are dedicated blocks under specific categories.</li> <li>If objects are already marked as secured no action is needed but if the object isn't marked as secured please change those and grant permission to the users accordingly</li> <li>Custom MDF objects with more than 2 levels of composites aren't supported.</li> <li>Each custom MDF object can be added only once.</li> </ul> |
| Miscellaneous Permissions            | <p>2 new permissions have been introduced to control the visibility of Pending workflow details:</p> <ul style="list-style-type: none"> <li><a href="#">View Pending Approvals</a></li> <li><a href="#">Review Pending Approvals</a></li> </ul>   | <p>These permissions are applicable only for the latest People Profile and have no impact in the legacy People Profile or Manage data page.</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p><b>Note</b></p> <p>SAP SuccessFactors Mobile does not display <i>Effective date</i> and <i>Pending Approvals</i> link.</p> </div>  |
| Configuration of MDF objects         | <p>As an administrator user, you must select only the object name for configuring any MDF object.</p>   | <p>In the legacy People Profile, administrator could select the Config UI or Screen ID. However, now it has to be configured on the <i>Default Screen</i> field in <a href="#">Admin Center &gt; Configure Object Definitions</a> page.</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p><b>Note</b></p> <p>The configured UI can also be used for the workflow approval page.</p> </div>                                       |

| MDF Changes in Latest People Profile             | Description  | Notes  |
|--|--|--|
| Administrator user can select only the Object ID | As an administrator user, you can select only object ID and not the specific object on the configuration screen. |  |
| Quick card for MDF objects                       | No support for Quick cards in latest People Profile for MDF.   | With no support for Quick card, you can't navigate to a specific object. If you want to see the details of such an object, use the Manage data page. |

## Related Information

[Full Profile of the Latest People Profile](#)  
[Configuring Categories and Cards in Full Profile](#)

# 8 Setting Up Security for Object Instances

## 8.1 Setting Up Security for MDF Using Role-Based Permissions (RBP)

### 8.1.1 Overview

This section describes how to add security based on RBP to the MDF object definition. You can add security at object, field, and association levels.

Setting up security allows you to control which users can view, edit, or import/export data in an object or field. With MDF you can set up security at the more granular field level. For objects and fields, you can control access based on roles within the system.

For more information about RBP and the allowable permission categories, see the following guides on SAP Help Portal:

- [Implementing Employee Central Core](#)
- [Using Role-Based Permissions](#)
- [Implementing Role-Based Permissions](#)

### Enabling Security

Ensure that you enabled security in the object. For more information, see [Adding Security \[page 38\]](#).

After an object is secured, you can control access to that object in role-based permissions.

Optionally, you can define which field in the object definition is used to determine the target population in role-based permissions. You can specify any user field in the current object definition as the *RBP Subject User Field*. The value of the specified *RBP Subject User Field* can then be used to create user-based permission groups in RBP. You can use any field in the current object definition of the type "user" as the *RBP Subject User Field*. It does not need to be the externalCode.

#### ⓘ Note

There are 2 Subject User Fields in MDF:

- Subject User (introduced first and primarily used by workflow)
- RBP Subject User (introduced later and used by MDF permissions)

This was done to ensure backward compatibility and flexibility. If the RBP subject user is not populated, then the system DOES NOT default to the first subject user ID.

MDF RBP have no dependency on the workflow subject user and only depend on the RBP subject user. If the RBP subject user is populated, it will take the RBP subject user. If the RBP subject user is not populated, the object will not be regarded as user-based in MDF RBP and it will support target criteria instead of target

population. This means, if the *RBP Subject User Field* is not specified in the object definition, then this object will be regarded as target criteria based instead of user based. If the *RBP Subject User Field* is defined in the definition, then this field must be required so it should always have a value to avoid cases where the RBP subject user value is not specified in object instance even though its included in the definition. If the customer doesn't want the RBP subject user, then the object definition must be changed.

## 8.1.2 Reporting

This sections informs you that you can run ad-hoc reports to see permissons.

Ad hoc reports for role-based permissions now include MDF objects. You can run an ad hoc report to view the MDF permissions for any combination of objects, users, groups, and roles. This allows you to conduct, for example, security audits.

For more information, see the *Role-Based Permissions Implementation Guide* from the SAP Help Portal at

► [Foundation](#) ► [Implementation Guides](#) ►.

## 8.1.3 Enabling Security for Objects

This section explains how to set up security for objects.

### Context

#### ⓘ Note

You can set up permissions for the end users for specific MDF objects. This ensures that they can only see the MDF entities that they're authorized to import.

After you have enabled security for an object, you can control permission to view, edit, import, and export data in that object, using role-based permissions.

#### ⓘ Note

To grant permissions on the field level, see [Enabling Security for Fields \[page 91\]](#).

### Procedure

1. Navigate to the Admin Center.
2. In the Tools search field, type *Manage Permission Roles*.

3. Choose a permission role.
4. On the [Permission Role Detail](#) page, click [Permission...](#)
5. Under [User Permissions](#), click [Miscellaneous Permissions](#).
6. To grant permissions to view, edit, or import/export data in that object, select the appropriate checkboxes. The permissions available depend on the object type.

| Object Type                        | Permissions Available   |
|------------------------------------|---|
| <b>Non-effective-dated objects</b> | View<br>Edit<br>Import<br>Export  |
| <b>Effective-dated objects</b>     | View Current<br>View History<br>Insert<br>Correct<br>Delete<br>Import<br>Export |
| <b>Composite child objects</b>     | View<br>Correct<br>Create<br>Adjust Order<br>Delete<br>Import<br>Export         |

7. Click [Done](#).

## 8.1.4 Enabling Security for Fields

This section explains how to control access to the individual fields of an object.

### Context

You can also set up security at the field level.

## Procedure

1. Open the *Permission Settings* dialog and select *Miscellaneous Permissions*, as described in [Enabling Security for Objects](#) [page 90].
2. Select the *Field Level Overrides* checkbox.
3. Choose a field from the dropdown and set the corresponding permission to *Read Only* or *No Access*. For example, this image shows field-level permissions of the *Position* object.

The screenshot shows the 'Permission settings' dialog box. On the left, a list of entities is shown, with 'Miscellaneous Permissions' selected. The main area displays settings for the 'Position' object. Under 'Field Level Overrides', a table lists fields and their permissions:

| Field          | Permission   |
|----------------|--------------|
| Change Reason  | No Access    |
| Position Title | Read Only    |
| No Selection   | No Selection |

Other settings for 'Position' include: Visibility:  View Current,  View History; Actions:  Create,  Insert,  Correct,  Delete,  Import/Export; and  Field Level Overrides. Below this, settings for 'Reporting Entity for North America' and 'SAP System Configuration' are also visible.

4. Repeat for any additional fields that require field-level permissions.
5. Click *Done*.

## 8.1.5 Enabling Security for Parent Object Associations

This section explains how to security for parent object associations.

## Procedure

1. Navigate to the Admin Center.
2. In the Tools search field, type *Manage Permission Roles*.

3. Choose a permission role.
4. On the [Permission Role Detail](#) page, click [Permission...](#)
5. Under [User Permissions](#), click [Miscellaneous Permissions](#).
6. To grant view or edit permissions to the MDF object, select the [View](#) and/or [Edit](#) checkboxes.
7. Select the [Field Level Overrides](#) checkbox.
8. Choose an association from the dropdown and set the corresponding permission to either [Read Only](#) or [No Access](#) for that field.
9. Click [Done](#).

## 8.1.6 Enabling Security for Composite Child Permissions

This section explains how to add security for parent object associations.

### Context

You can specify the field level override and object level permission in the applicable composite for composite children because valid-when relationships are independent.

#### Note

If a composite is a user-based object, it won't be shown on the RBP setting page.

### Procedure

1. Secure the composite child object definition. For more information, see [Enabling Security for Objects \[page 90\]](#).
2. Navigate to the Admin Center.
3. In the Tools search field, type [Manage Permission Roles](#).
4. Choose a permission role.
5. On the [Permission Role Detail](#) page, click [Permission...](#)
6. Under [User Permissions](#), click [Miscellaneous Permissions](#).
7. Configure the [View](#), [Correct](#), [Create](#), [Adjust Order](#), and [Delete](#) permissions as needed.
8. Select the [Field Level Overrides](#) checkbox.
9. Choose the composite child object field from the dropdown and set the permission.
10. Click [Done](#).



## 8.1.7 Order of Permissions

This section explains the order in which the MDF checks the permissions of the parent object.

The availability of permission settings for fields, associations, and child objects depends on the security settings of the parent object. If you configured both the parent and composite child permissions when you created an instance of the object, MDF checks the permissions in the following order:

1. Parent
2. Association Relationship
3. Child

| If the Parent Object Permission Is... | Then the Permission for...  |
|---------------------------------------|---|
| Editable                              | Association relationship is editable, read only, no access<br>Composite object is none, view, correct, insert, delete, adjust order |
| Read only                             | Composite object is none, view<br>Association relationship is read only, no access  |

## 8.2 Restricting Access

### 8.2.1 Assigning Roles to Groups

After creating your roles, you must assign the role to a group of employees. This ensures that employees are given access the permissions they need to perform their tasks.

#### Procedure

1. Go to the [Admin Center](#).
2. In the [Tools Search](#), search for **Manage Permission Roles**.
3. Select one of the permission roles you created.
4. In the [Grant this role to](#) section of the [Permission Detail](#) screen, choose [Add](#).


#### ⓘ Note

If you have enabled Onboarding or Learning, you have an additional option [Add For External Target Population](#) where you can restrict data access by defining external target populations.

5. When the [Grant this role to](#) screen displays, select [Permission Group](#).

**Grant this role to...**

**1: Define whom you want to grant this role permission to.** ?



Grant role to:

Permission Group... ▼

None Selected

Select...

Allow their manager to have the same permission access. ?

1 ▼ level(s) up (for example: Direct manager is 1 level up)

- Click [Select](#) to select the access groups you wish to assign to this permission role.

You can allow managers to have the same permissions and define how many levels up in the hierarchy you want this permission to be granted. However, allowing respective managers to have the same permissions may have a negative impact on the performance. The hierarchy then has to be checked whenever such a manager tries to access an element that was permissioned this way.

- Exclude Granted Users:**

For some permissions, it might be necessary to exclude the granted users from applying the permissions on themselves. For this, select [Exclude Granted User from having the permission access to themselves](#).

#### Example

If the role grants permission to edit the salary, you want to prevent the members of this permission group to be able to edit their own salary as well.

- Click [Done](#) to assign this role to the defined users. You're taken back to the [Permission Role Detail](#) page.
- Click [Save Changes](#) to complete creating the role.

## Next Steps

If required, assign a target population to your role.

## 8.2.2 Restricting Data Access of a Role with Target Population or Criteria

You can restrict data access by defining target populations or criteria of roles that require tasks to be performed on behalf of another employee.

### Context

Target populations and criteria allow you to give employees such as managers and administrators access to data or tasks that need to be maintained for other employees. Not all permissions require you to define a target population.

For example, if the permission includes just the access to an application (such as the Learning Access Permission), there's no need to add a target user group or target criteria.

For most of the permissions and secured MDF object definitions, you can restrict data access by either setting target population or adding target criteria. For a few of the secured MDF object definitions, you can set its target population and further restricting the target population with additional target criteria.

#### → Remember

Only object definition *Employee Time* can be configured with concurrent target population and target criteria.

You can identify permissions and object definitions that are available for target population configuration, in the *Permission settings* screen, by the "t" icon next to the permission name with the following text displayed: **t= Target needs to be defined.**

## Procedure

1. Go to the *Admin Center*.
2. In the *Tools Search*, search for **Manage Permission Roles**.
3. Select one of the permission roles you created.

#### ⓘ Note

As of the 2H 2022 release, you can configure target criteria for a permission role whose target population is external users. If you have existing permission roles granted to groups whose target population is external users, review the target criteria configurations of those roles. The target criteria configurations for said permission roles will stay as null if you don't open and save the roles. When you open and save the roles, the target criteria configurations are set to all.

Please note that this feature isn't available for instances using Onboarding 1.0.

4. Choose *Edit Granting* for a role you've assigned this permission to in the *Grant this role to* section of the *Permission Detail* screen.
5. Define target population in the second section of the *Grant this role to...* popup window.  
To exclude the granted users from applying the permissions on themselves, select *Exclude Granted User from having the permission access to themselves*.
6. Define target criteria in the third section of the *Grant this role to...* popup window.

#### ⓘ Note

Only secured MDF object definitions whose *RBP Subject User Field* is left empty are available for target criteria configuration.

7. Click *Done* to assign this role to the defined users. You're taken back to the *Permission Role Detail* page.
8. Click *Save Changes* to complete creating the role.

## 8.2.3 Enabling Search Field Restriction

MDF allows you to restrict the list of values displayed for a Generic Object(GO) field based on logged-in user's target population. For example, if a user only has permission to view departments in the Engineering group, the dropdown displays the values accordingly.

### Context

For better understanding, let's take a look at the following scenario, where we have a secured and non-secured MDF object and generic object fields A and B:

Generic Object Field Drop-down Behavior

| Scenario                                     | Enable RBP Target Criteria for Value Help (Only with data type Generic Object) | Generic Object Field | User Permissions                                  | Drop-down Behavior for Reference Field or Valid When Association        |
|--|--|----------------------|---|---|
| Secured MDF Object                           | Enabled  | A                    | Disabled  | Values are not visible.   |
| Secured MDF Object                           | Enabled  | A                    | Enabled   | Values are visible if the role-based permission is enabled.             |
| Non-secured MDF Object                       | Enabled  | B                    | Enabled ( <i>Access to non-secured objects</i> )  | Displays values only for generic objects (GO) you're allowed to access. |
| Non-secured MDF Object                       | Enabled  | B                    | Disabled ( <i>Access to non-secured objects</i> ) | Values are not visible.   |
| Secured MDF Object or Non-secured MDF Object | Disabled   | A or B               | Not applicable                                    | All values are visible.   |

### Procedure

1. Log on to Provisioning and select the company to work with.

#### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

2. In **Edit Company Settings**, select **Company Settings**.
3. Enable the *Enable RBP Target Criteria for Value Help (Only with data type Generic Object)* option.
4. Click **Save**.

## Next Steps

Proxy as another user to validate that search criteria has been restricted on **Manage Data** page or on **Employee Profile Portlet** based on that user's permission.

## 9 MDF Picklists

A picklist is a dropdown that allows you to restrict the values that can be selected for a field. For example, you can define a picklist to show marital status with fixed set of values like Married, Single, and Divorced.

You can now manage your picklists using the [Picklist Center](#). All picklists including legacy picklists that haven't been migrated can be accessed from the Picklist Center. You can no longer access and manage picklists from the [Configure Object Definitions](#) page from Q4 2019 release.

To overcome the issues associated with managing legacy picklists, all existing picklists are being migrated to MDF. When the migration is complete, you'll be able to manage all picklists using Picklist Center. The Picklist Center aims to reduce the risk of errors and potential for data corruption.

### Note

You can no longer perform any operation on legacy picklists if they are not migrated to MDF. Ensure that you have migrated to MDF picklists to perform operations on picklists.

### Granting Permissions to Access Picklists

Before you start managing your picklists, you must grant users, permissions essential to creating, updating, and managing picklists. Typically, administrators manage picklists. You can authorize a single admin or a group of admins using role-based permissions.

To access Picklists Management and Picklist Center ensure that the following checkboxes are enabled.

- [Picklist Management and Picklists Mappings Set Up](#) under [Manage System Properties](#)
- [Configure Object Definitions](#) under [Metadata Framework](#)

### Migrated Picklists

Picklists migrated to MDF can be edited using [Picklist Center](#) or [Extension Center](#). It's important to note that the **Status** field of the picklist assumes different values post migration.

The **Status** field for legacy picklists before migration can assume these values:

- **Active:** The picklist value is available for use.
- **Deleted:** Disables the picklist value from the system (it isn't displayed on the UI anymore). It's important that you don't delete the entry. Use the Deleted and Obsoleted states, instead.
- **Obsoleted:** Disables the picklist value from being selected in new user records. Continues to appear in the Query tool for older records.

Active legacy picklist values remain active in MDF picklists, while obsoleted and deleted picklists values are migrated as Inactive MDF picklist status value.

## 9.1 Searching for a Picklist

By default, you can search for a picklist using its external code or the picklist's name. However, you can also search for a picklist using its display order, status, and so on.

### Prerequisites

Ensure you have either one of these permission to view and edit the relevant picklist objects in MDF:

- [Metadata Framework](#) > [Configure Object Definitions](#) >
- [Picklist Management and Picklists Mappings Set Up](#)

### Context

On the [Picklist Center](#) page, you can search, create, sort, delete, and filter picklists. You can also filter your picklists by their effective date and define the columns to be displayed on the page. Icons to perform these operations are displayed prominently against the [Picklists](#) table.

### Procedure

1. To search for your migrated MDF picklists, go to [Admin Center](#) > [Picklist Center](#).
2. In the [Search](#) field, enter the external code of the picklist, the picklist's name, or Legacy Picklist ID of the picklist. You can also choose the effective date in the [As of date](#) field, to further filter the picklists to help you located your picklist.

The picklist is displayed.

## 9.2 Adding Picklists to a Bundle for Transportation

Add a picklist to a bundle so that you can transport the picklist to a paired system using Configuration Transport Center.

### Prerequisites


- You've created a transport bundle in Configuration Transport Center.

- You have the necessary permission to create, update, and manage picklists.
- To access *Picklist Center*, you have the ► *Administrator Permissions* ► *Metadata Framework* ► *Configure Object Definitions* ► permission.
- You have the ► *Administrator Permissions* ► *Configuration Transport Center* ► *Access to transport configurations* ► permission.

## Context

Bundles are artifacts in Configuration Transport Center that contain configurations of your system. You can use bundles to transport the configuration of a source system to a paired target system so that you don't need to manually configure it.

## Procedure

1. Go to ► *Admin Center* ► *Picklist Center* ►.
2. In the *Picklist Search* field, enter the external code or name of the picklist and select the picklist that you want to transport.
3. The *Add to Transport Bundle* option is available in the *Picklist Center* overview page and the *Picklist Details* page. You can select the *Add to Transport Bundle* option to add the current picklist to an existing bundle.
  - a. Choose *Add to Transport Bundle*  icon in the *Picklist Center* overview page to add the selected picklist to an existing bundle. The *Add to Transport Bundle* icon will be enabled only if you select a picklist.
  - b. Choose an existing picklist to navigate to the *Picklist Details* page. Select the *Add to Transport Bundle* option at the top right of the page to add a picklist to transport bundle.

A list of available transport bundles appears.

### Note

Adding a picklist to a transport bundle transfers the complete picklist, including all timeslices and picklist values. There's no support for a single timeslice or few picklist values individually.

4. Select the bundle that you want to add the configuration to and choose *Save*.

## Results

The picklist is added to the transport bundle.



## 9.3 Creating a Picklist

Create an MDF picklist using the Picklist Center.

### Prerequisites

Ensure you have either one of these permission to view and edit the relevant picklist objects in MDF:

- [Metadata Framework > Configure Object Definitions](#)
- [Picklist Management and Picklists Mappings Set Up](#)

### Context

Since picklist values are entered individually, one after another, it takes time to create a large picklist from scratch.

To add a large number of picklist values at once, we recommend using [Admin Center > Import and Export Data](#).

#### Note

Adding custom fields, rules, or setting Picklist as RBP secured is not supported. In future releases, we will impose these restrictions. If you have these configurations, it might lead to issues in the application. Hence, remove them.

#### Remember

When you have cascading picklists, always create the parent picklist before creating the child picklist. This is required to map the parent and child picklist correctly.

### Procedure

1. Go to [Admin Center > Picklist Center](#).
2. Click the + icon, to create a new picklist.  
The [Create a New Picklist](#) window is displayed on the UI.
3. Enter the picklist data.

| Field                       | Description   |
|-----------------------------|---|
| <b>Code</b>                 | Unique business key that is used to identify the picklist. (If you're familiar with legacy picklists, these are the equivalent of the picklist ID.)   |
| <b>Name</b>                 | Name of the picklist. For example, if the picklist is for the type of address (home, mailing, business, billing, and so on), you could name the picklist "Address Type."  |
| <b>Status</b>               | Specifies whether the picklist is active or inactive, that is, whether it's available for use.  |
| <b>Effective Start Date</b> | A picklist can have different sets of picklist values for different start dates.  |
|                             | <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p><b>❖ Example</b></p> <p>To add a new value <b>New York</b> to the <i>Location</i> picklist but you want this picklist to be available for this location after January 1, 2017. Enter in <b>01/01/2017</b> as the effective date, this picklist that contains the new value is available after the January date.</p> </div> |
| <b>Display Order</b>        | Order in which the picklist values are listed in the picklist dropdown in the application where it is used. The options available currently are alphabetical, numerical, and custom sort.   |
| <b>Parent Picklist</b>      | Name of the parent picklist that has been selected, in a scenario where cascading picklists are used.   |
| <b>Legacy Picklist ID</b>   | If you want the picklist to be used in non-MDF scenarios, enter the legacy picklist ID. Note that the value is case-sensitive and must be unique across all picklists. If no value is provided, the code value is used as the default value.  |

A new picklist is created.

- Use the add icon **+** to create a new picklist value.

The *Add New Value* window is displayed.

- Enter picklist values.

For example, if your picklist is State, your values could be Alabama, Alaska, California, and so on.

| Field                | Description   |
|----------------------|---|
| <b>External Code</b> | MDF external code. A unique business key that is used to identify the picklist value, for example, state_Alabama. |

| Field                           | Description   |
|---------------------------------|---|
| <b>Label</b>                    | For example, if the external code is state_Alabama, the label would be Alabama. You can also click on the Globe icon to add the translated version of the label value.  |
| <b>Status</b>                   | Specifies whether the picklist value is active or inactive, that is, whether it's available for use.  |
| <b>Sort Order (L Value)</b>     | If you chose the display order as <b>Custom</b> , you can enter a numeric value here to help you define the sort order of the picklist values.  |
| <b>Parent Picklist Value</b>    | <p>The value of the parent picklist that causes the picklist value to appear.</p> <p>For example, the picklist value Alabama would have a parent picklist value United States. The picklist value Alberta would have a parent picklist value Canada.</p>  |
| <b>Non Unique External Code</b> | <p>Legacy external code, only used by migrated legacy picklists. Unlike the MDF external code, it can be blank or duplicated.</p> <div data-bbox="841 940 1429 1245" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Note</b></p> <p>Although this information is optional, it is recommended that you fill this out. When you create a new picklist value, the application automatically copies the value of the external code to this field.</p> <p>In some instances, if this field is left blank, certain application pages may result in errors.</p> </div>   |
| <b>Min Value</b>                | <p>Enter the minimum numeric value for the picklist value.</p> <p>Min value and max value are used to define the range of numerical values associated with each picklist option. These fields are typically utilized when you want to assign numeric values to the picklist options for sorting or comparison purposes.</p> <div data-bbox="841 1514 1429 1869" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Example</b></p> <p>If you have a picklist for employee performance ratings with options like "Exceeds Expectations," "Meets Expectations," and "Needs Improvement," you might assign numeric values to them using the "Min Value" and "Max Value" fields. "Exceeds Expectations" could have a higher range than "Meets Expectations" and so on. This allows for numerical comparisons in reports or other system functionalities.</p> </div> |

| Field     | Description   |
|-----------|---|
| Max Value | Enter the maximum numeric value for the picklist value. |

#### Note

- The two grayed-out fields *R Value* and *Value* are only enabled for migrated legacy picklists.
- When legacy picklists are migrated to MDF picklists, an `optionId` is generated for existing MDF picklist values. For legacy picklist option that has been migrated, the existing `optionId` is retained. After the migration, when you create a picklist from the *Picklist Center*, an option ID is generated for all records. The present effective dated record is used; other effective dated entries for the parent picklist aren't considered. The system-generated ID is not editable and cannot be retained when you delete the picklist and create a new one with the same external code.

6. Click *Create* to add a picklist value to the picklist.

The picklist values are now created.

## 9.4 Editing a Picklist

Edit or update an MDF picklist using Picklist Center.

### Prerequisites

Ensure you have either one of these permission to view and edit the relevant picklist objects in MDF:

- [Metadata Framework > Configure Object Definitions](#)
- [Picklist Management and Picklists Mappings Set Up](#)

### Context

You can use Picklist Center to edit all fields in a picklist, from changing the external code to attaching a new parent picklist.

#### Note

If a picklist is referenced elsewhere, editing its external code could lead to data loss on MDF objects where it has been used.

## Procedure

1. Go to ► [Admin Center](#) ► [Picklist Center](#) ►.
2. Search for the picklist you want to edit and click on the selected picklist's effective start date to start editing picklist.
3. Click [Edit](#) in the [Picklist Details](#) page to modify the picklist.

### Note

- You can now edit legacy picklist IDs from Picklist Center after they're migrated to MDF. The Edit Picklist page allows you to edit **Legacy Picklist ID** field. Please note that a legacy picklist ID must be unique across all picklists.  
For picklists that are configuration outside MDF, for example, Succession Data Model, the value of Legacy Picklist ID needs to be used. Ensure that any changes made to the Legacy picklist ID are adjusted in all references to this picklist outside MDF.
- Since MDF picklists are effective-dated, you can always control the date from which the record takes effect.

4. Click [Save](#).

## 9.5 Picklist Values

You can add values to your picklist.

For example, if you create a new department, you can add new values such as Human Resources, IT, Finance, and so on, as values to your picklist. The list of these values is displayed in the picklist dropdown when they're used in a different place within the system.

### [Creating and Editing a Picklist Value \[page 106\]](#)

Create a new picklist value or edit an existing picklist value, using Picklist Center.

### [Sort Order for Picklist Values \[page 108\]](#)

You can choose the sort order for picklist values that are listed in the picklist dropdown.

### [Deactivating Picklist Values \[page 108\]](#)

Deactivate a picklist value from a picklist, using Picklist Center, when it's no longer required.

### 9.5.1 Creating and Editing a Picklist Value

Create a new picklist value or edit an existing picklist value, using Picklist Center.

## Prerequisites

Ensure you have either one of these permission to view and edit the relevant picklist objects in MDF:

- [► Metadata Framework](#) > [Configure Object Definitions](#) >
- [Picklist Management and Picklists Mappings Set Up](#)

## Context

### ⓘ Note

Adding custom fields, rules, or setting Picklist Value as RBP secured is not supported. In future releases, we will impose these restrictions. If you have these configurations, it might lead to issues in the application. Hence, remove them.

## Procedure

1. Go to [► Admin Center](#) > [Picklist Center](#) >.
2. Search for the picklist you want to edit and click on the selected picklist's effective start date to start editing picklist.

The [Versions](#) page along with its picklist values are displayed.

3. Add or edit a picklist value.
  - Use the add icon **+** to create a new value.
  - Click an existing value to edit it.

Based on the action you chose, you are directed to either the [Add New Value](#) page or the [Edit Picklist Value](#) page.

4. Enter picklist value data, as needed.

### ⓘ Note

The two grayed-out fields, [R Value](#) and [Value](#) are only enabled for migrated legacy picklists.

5. Click [Save](#).

Your new picklist value is either created or edited.

**Task overview:** [Picklist Values \[page 106\]](#)

## Related Information

[Sort Order for Picklist Values \[page 108\]](#)

[Deactivating Picklist Values \[page 108\]](#)

## 9.5.2 Sort Order for Picklist Values

You can choose the sort order for picklist values that are listed in the picklist dropdown.

Picklist values can be sorted in three specific ways. By default, the sorting order is set to *Alphabetical*. However, you can change the display order to *Numeric* or *Custom*.

While you choose the order of your picklist values while creating your picklist in the *Picklist Center*, you won't typically see the manner in which your picklist values are displayed. The values are seen in the order you've chosen when your picklist is being used in a different place within the system.

For example, if you're entering employee details and choosing the state the employee is from, you can create a picklist called **United States**, with the picklist values being all the states within the country. If you have chosen the display order as **Alphabetical**, the admin notices that the states are listed in alphabetical order in the picklist dropdown.

The three kinds of display orders are:

- *Alphabetical*. The picklist values are displayed in alphabetical order.
- *Numeric*. The picklist values are displayed in numeric sequence.
- *Custom*. You can choose a numeric sort order value that defines how the picklist values are displayed in the picklist dropdown.

### Note

You'll see the *Sort Order (L Value)* field when you create or edit a picklist value **only** if you chose *Custom* as the display order while creating your picklist.

Parent topic: [Picklist Values \[page 106\]](#)

## Related Information

[Creating and Editing a Picklist Value \[page 106\]](#)

[Deactivating Picklist Values \[page 108\]](#)

## 9.5.3 Deactivating Picklist Values

Deactivate a picklist value from a picklist, using Picklist Center, when it's no longer required.

## Prerequisites

Ensure you have either one of these permission to view and edit the relevant picklist objects in MDF:

- [Metadata Framework](#) > [Configure Object Definitions](#) >
- [Picklist Management and Picklists Mappings Set Up](#)

## Context


You can select picklist values for deactivation by choosing the checkbox against each picklist value.

## Procedure

1. Go to [Admin Center](#) > [Picklist Center](#) >
2. In the [Picklist Search](#) field, enter the external code or name of the picklist. Choose the picklist from which you want to remove picklist values.

### Note

You can also click on the effective start date of the picklist to view the picklist values associated to the picklist.

- You're directed to the [Versions](#) page of the picklist to make any edits to it.
3. Choose the picklist values that you want to deactivate and click  .  
A warning message is displayed letting you know of the consequences of your action.
  4. Click [Yes](#) to continue.  
Your picklist value is now deactivated.

**Task overview:** [Picklist Values \[page 106\]](#)

## Related Information

[Creating and Editing a Picklist Value \[page 106\]](#)

[Sort Order for Picklist Values \[page 108\]](#)



## 9.6 Picklist Versions

MDF picklists are effective-dated. To make changes to the picklist values for a different date, create a new version of a picklist with a different effective start date, and then, modify your picklist values. For more information, see **Inserting a New Picklist Version**.

You can view any version of the selected picklist on the [Versions](#) page. In addition to the version of the picklist, you can also view the status of the version, its effective start date, when it was last modified, and details of the user who modified the selected picklist version.

### Note

Picklists that are configured outside MDF are not effective dated, for example, JobInformation. In such picklists, when they have multiple versions, the current date is considered as the effective date. This behavior is similar to the picklist in MDF objects that have effective dating set as None.

### To view picklist versions

1. Go to the [Picklist Center](#) and search for the picklist you want to view.
2. Click on the external code of the picklist to be directed to the [Versions](#) page. All the versions associated with the picklist are listed here.
3. Click on the picklist version you want to work with.

### Related Information

[Creating a Picklist Version \[page 110\]](#)

## 9.6.1 Creating a Picklist Version

You can add new version for a selected picklist as required by your company.

### Context

You can only add one version for a picklist at a time. You can't insert a time slice earlier than the initial record and you can't insert a time slice for an existing version with the same effective start date.

## Procedure

1. Go to the [Picklist Center](#) and search for the picklist you want to view.

### Note

To manage legacy picklists, use the legacy **Picklist Management** to export and import them as a CSV file.  
In either situation, if you want to work offline with your picklist, use **MDF Import and Export** page.

2. Click on the external code of the picklist you want to create a version for.

You're directed to the [Versions](#) page. All the versions associated with the picklist are listed here.

3. Click + to add a version.
4. Select the date using date picker option from the [Add New Version](#) dialog.

By default, current date is selected on the [Add New Version](#) dialog.

5. Click *OK*.

## Results

The new version for the picklist gets added in the [Versions](#) page. When you insert a version between the existing versions of the picklist, all the picklist values and the header information that are available for the previous versions are carried over to the inserted version.

## 9.6.2 Deactivating a Picklist Version

You can deactivate a version from the picklist when it's no longer required.

## Context

Deactivated picklist versions aren't available for use in the application.

## Procedure

1. Go to the [Picklist Center](#) and search for the picklist you want to view.

### Note

To manage legacy picklists, use the legacy **Picklist Management** to export and import them as a CSV file.  
In either situation, if you want to work offline with your picklist, use **MDF Import and Export** page.

2. Click on the external code of the picklist.

You're directed to the [Versions](#) page. All the versions associated with the picklist are listed.

3. Select the version of the picklist that you want to deactivate and click [Deactivate](#).

A warning message is displayed letting you know of the consequences of your action.

4. Click [Yes](#) to deactivate the version.

## Results

The selected picklist version gets deactivated. If picklist has a reference, an warning message is displayed asking you to confirm your decision to deactivate it.

## 9.7 Deactivating a Picklist

If picklist has only one version, then you can deactivate an entire picklist. Deactivated picklists aren't available for use in the application.

### Context

You can select the picklist you want to deactivate by choosing the checkbox against each picklist.

### Procedure

1. Go to [Admin Center](#) > [Picklist Center](#).
2. In the [Picklist Search](#) field, enter the external code or name of the picklist and select the picklist that you want to deactivate.

You can also click on the effective start date of the picklist to view picklist data.

You're directed to the [Versions](#) page of the picklist to make any edits to it.

3. Click  to deactivate the picklist.

A warning message is displayed letting you know of the consequences of your action.

4. Click [Yes](#) to continue.

## 9.8 Cascading Picklists

The `Picklist` and the `PicklistValue` objects have a cascading property, that is, they have a parent-child relationship.

To use cascading relationship, you have to add field criteria based on the picklist used. Using cascading picklists you can limit the values of a picklist to depend on the selection of another picklist. For example, if you want to create a cascading property between three picklists, **Country**, **State**, and **City**.

The value of **State** picklist depends on the selected **Country**. And the value of **City** picklist depends on the selected **State**. You can select the **Country** as India or the USA. Based on this selection, values displayed in State vary. Similarly, based on the selected **State**, the list of cities varies. If cascading property isn't configured, then you can configure **Country** as India, **State** as California, and **City** as Bangalore.

To create a cascading property between picklists, **Country**, **State**, and **City**, perform the following configurations:

1. In *Admin Center > Picklist Center*, create three picklists **Country**, **State**, and **City**. Ensure to set *Parent Picklist* value of picklist **State** to **Country** and *Parent Picklist* value of picklist **City** to **State**.
2. In *Admin Center > Configure Object Definitions*, create a Generic Object of type `cust_CascadingPicklistTest` that uses all three picklists.

The screenshot shows the configuration page for the object `cust_cascadingpicklist`. The configuration options include:

- Code: `cust_cascadingpicklist`
- Effective Dating: Basic
- API Visibility: Not Visible
- Status: Active
- MDF Version History: No Selection
- Default Screen: (empty)
- Label: `cust_cascadingpicklist`
- Description: (empty)
- Enable Full Text Enhanced Search Indexing: No
- API Sub Version: `V1.1`
- Subject User Field: (empty)
- Workflow Routing: (empty)
- Pending Data: No
- Todo Category: `Generic Object Change Requests`

The Fields table is as follows:

| Name                                     | Database Field Name             | Maximum Length | Data Type | (22) More |
|--|---------------------------------|----------------|-----------|-----------|
| <code>externalCode</code>                | <code>externalCode</code>       | 128            | String    | Details   |
| <code>externalName</code>                | <code>externalName</code>       | 128            | String    | Details   |
| <code>cust_Country</code>                | <code>sfFields.sfField1</code>  | 38             | Picklist  | Details   |
| <code>cust_State</code>                  | <code>sfFields.sfField2</code>  | 38             | Picklist  | Details   |
| <code>cust_City</code>                   | <code>sfFields.sfField3</code>  | 38             | Picklist  | Details   |
| <code>mdfSystemEffectiveStartDate</code> | <code>effectiveStartDate</code> | 10             | Date      | Details   |

3. To configure the cascading property, update the Field Criteria of `cust_State` and `cust_City` as:
  1. Open `cust_State` details, update Field Criteria `parentPickListValue` to `cust_Country`.

Obj Details

Name **cust\_State**

Database Field Name sfFields.sfField2

Maximum Length 38

Data Type Picklist

Valid Values Source MyState

Hide Old Value No

Decimal Precision

Include Inactive Users No

UI Field Renderer

Transient No

Help Text

Private or Sensitive Information No

Show Trailing Zeros No

Default Value

Hide Seconds No

End Of Period No

BE Column Name

Required No

Visibility Editable

Status Active

Label State ?

Cascade None

Inactivated By No Selection

Rules

External Code

Field Criteria

| Source Field Name          | Destination Field Name      | Default Destination Value | Status |
|----------------------------|-----------------------------|---------------------------|--------|
| <b>parentPickListValue</b> | <b>cust_Country</b>         |                           | Active |
| parent.effectiveStartDate  | mdfSystemEffectiveStartDate | 1900-01-01                | Active |

Done

- Open **cust\_City** details, update Field Criteria **parentPickListValue** to **cust\_State**.

## Details

|                                  |                   |
|----------------------------------|-------------------|
| Name                             | cust_City         |
| Database Field Name              | sfFields.sfField3 |
| Maximum Length                   | 38                |
| Data Type                        | Picklist          |
| Valid Values Source              | MyCity            |
| Hide Old Value                   | No                |
| Decimal Precision                |                   |
| Include Inactive Users           | No                |
| UI Field Renderer                |                   |
| Transient                        | No                |
| Help Text                        |                   |
| Private or Sensitive Information | No                |
| Show Trailing Zeros              | No                |
| Default Value                    |                   |
| Hide Seconds                     | No                |
| End Of Period                    | No                |
| BE Column Name                   |                   |
| Required                         | No                |
| Visibility                       | Editable          |
| Status                           | Active            |
| Label                            | City              |
| Cascade                          | None              |
| Inactivated By                   | No Selection      |

Rules

External Code

Field Criteria

| Source Field Name         | Destination Field Name      | Default Destination Value | Status |
|---------------------------|-----------------------------|---------------------------|--------|
| parentPickListValue       | cust_State                  |                           | Active |
| parent.effectiveStartDate | mdfSystemEffectiveStartDate | 1900-01-01                | Active |

[Done](#)

4. Go to *Admin Center > Manage Data*, from Create New list, select **cust\_CascadingPicklistTest**.
  1. Select **India** from the Country list. Now check the values in State list. Only states from India are listed.

cust\_cascadingpicklist: Division\_IND (Division)

\* externalCode

externalName

Country

State

City

\* Effective Start Date

2. Select a state. Now, check the values in the City list. Only cities that are related to the state you have select are listed.

cust\_cascadingpicklist: Division\_IND (Division)

\* externalCode

externalName

Country

State

City

\* Effective Start Date

cust\_cascadingpicklist: Division\_IND (Division)

\* externalCode

externalName

Country

State

City

\* Effective Start Date

## 9.9 Deleting Picklists

Deleting picklists, picklist versions, or picklist values results in the loss of data references within or outside MDF. A deleted picklist, picklist version, or picklist value can't be restored.

SAP recommends not to delete picklists, picklist versions, or picklist values.

If picklists are deleted, it can't be recovered. Hence, you must only deactivate a picklist, picklist version, or picklist value.

### Related Information

[Deleting Picklist or Picklist Versions \[page 117\]](#)

[Deleting Picklist Values \[page 118\]](#)

## 9.9.1 Deleting Picklist or Picklist Versions

You can delete a picklist or a picklist version when it's no longer required.

### Context

SAP recommends not to delete picklists or picklist versions. If picklists are deleted, it can't be recovered. Hence, you must only deactivate a picklist, picklist version, or picklist value.

### Procedure

1. Go to ► [Admin Center](#) ► [Import and Export Data](#) ►.
2. From the *Select the action to perform* dropdown, select **Export Data**.

Fields corresponding to **Export Data** appears

3. From the *Select Generic Object* dropdown, select a picklist.
4. From the *Include Dependencies* dropdown, select **No**.
5. From the *Select all data records* dropdown, select **No**.

On selecting **No**, a *Select Objects* dropdown appears and here you can specify the specific picklists that you want to export.

6. Click *Export*.

An export job is triggered. You can track the status of this job in [Scheduled Job Manager](#).

When the job is completed, you can download the picklist package using the **Download Status** link in [View Details](#).

7. Open the CSV file. Retain the rows corresponding to the picklist or picklist version that you want to delete and remove the other rows from the CSV file.
8. In the rows corresponding to the picklist or picklist version that you want to delete, update the **Operations** column entries to **DELETE**.
9. Go to ► [Admin Center](#) ► [Import and Export Data](#) ►.
10. Select *Import Data* from the *Select the action to perform* dropdown list.

Fields corresponding to *Import Data* appears and by default *CSV File* tab is selected.

11. Select a picklist from the *Select Generic Object* dropdown list.
12. Click *Choose File* to select the CSV file to import in the *File* field.
13. Select *Incremental Load* from the *Purge Type* dropdown list.
14. Click *Import* and check the import status in [Scheduled Job Manager](#).



## 9.9.2 Deleting Picklist Values

You can delete a picklist value when it's no longer required.

### Context

SAP recommends not to delete picklist values. If picklists are deleted, it can't be recovered. Hence, you must only deactivate a picklist, picklist version, or picklist value.

### Procedure

1. Go to ► [Admin Center](#) ► [Import and Export Data](#) ►.
2. From the *Select the action to perform* dropdown, select **Export Data**.

Fields corresponding to **Export Data** appears

3. From the *Select Generic Object* dropdown, select *Picklist-Values*.
4. Click *Export*.

An export job is triggered. You can track the status of this job in *Scheduled Job Manager*.

When the status of this job is Completed, you can download the picklist package using the **Download Status** link.

5. Open the CSV file. Retain the rows corresponding to the picklist values you want to delete and remove the other rows from the CSV file.
6. In the rows corresponding to the picklist values that you want to delete, update the **Operator** column entries to **DELIMIT**.
7. Go to ► [Admin Center](#) ► [Import and Export Data](#) ►.
8. Select *Import Data* from the *Select the action to perform* dropdown list.

Fields corresponding to *Import Data* appears and by default *CSV File* tab is selected.

9. Select *Picklist-Values* from the *Select Generic Object* dropdown list.
10. Select *Incremental Load* from the *Purge Type* dropdown list.
11. Click *Choose File* to select the CSV file to import in the *File* field.
12. Click *Import* and check the import status in *Scheduled Job Manager*.

## 9.10 Importing and Exporting Picklists

### 9.10.1 Exporting an MDF Picklist

Exporting picklists allow you to migrate the picklists from one instance to another with the existing configurations.

#### Procedure

1. Go to [Admin Center](#) > [Import and Export Data](#).
2. From the *Select the action to perform* dropdown, select **Export Data**.  
Fields corresponding to **Export Data** appears.
3. Configure these parameters as required:
  - a. From the *Select Generic Object* dropdown, select a picklist that you want to export.
  - b. From the *Include Dependencies* dropdown, select *Yes* to include the dependencies. Else, select *No* to exclude dependencies.
  - c. From the *Include Immutable IDs* dropdown, select *Yes* to export Entity ID and Record ID. Select *No* to exclude exporting Entity ID and Record ID.
  - d. From the *Exclude reference objects* dropdown, select *No* to export the reference objects. Select *Yes* to include reference objects.
  - e. From the *Select all data records* dropdown, select *Yes* to export all the picklists. If you select *No*, a *Select Objects* dropdown appears and here you can select the specific picklists that you want to export.
4. Click *Export*.

An export job is triggered. You can track the status of this job in [Admin Center](#) > [Scheduled Job Manager](#).

#### Results

When the job is completed, you can download the picklist package using the *Download Status* link in *View Details*.

### 9.10.2 MDF Picklist Import

Similar to export, you can also import your picklist. Once you are done making changes to the exported picklist, you can import it back. You can import a picklist in two ways - CSV import and ZIP import.

Select the type of import relevant to your situation:

- Use CSV import if you want to make minor changes like - updating existing values, changing a label, adding values to the picklist.
- Use ZIP import if you want to do full purge, change the picklist headers and values separately.

### [Using CSV Import with MDF Picklists \[page 120\]](#)

Use CSV import to make minor changes like - updating existing values, changing a label, adding values to the picklist.

### [Using ZIP Import with MDF Picklists \[page 121\]](#)

Use ZIP import if you want to do full purge, change the picklist headers and values separately.

## 9.10.2.1 Using CSV Import with MDF Picklists

Use CSV import to make minor changes like - updating existing values, changing a label, adding values to the picklist.

### Procedure

1. Go to **Admin Center** > **Import and Export Data**.
2. Select **Import Data** from the *Select the action to perform* dropdown list.  
Fields corresponding to **Import Data** appears and by default **CSV File** tab is selected.
3. Select a picklist from the *Select Generic Object* dropdown list.
4. Click **Choose File** to select the CSV file to import in the *File* field.
5. Specify the file encoding from the *File Encoding* dropdown list.
6. Select **Incremental Load** from the *Purge Type* dropdown list.

#### Note

You aren't allowed to use **Full Purge** in CSV format as it leads to data loss. For example, if a picklist is referenced by an object, then after the full purge the reference goes off.

7. Select **Yes** to remove redundant effective-dated records from the *Suppress Redundant Date-Effective Records* dropdown list. Else, select **No**.
8. Choose **Business Key** or **External Code** as the *Key preference*.  
Make sure that you select the same key preference as the one you chose when you exported the picklist.
9. Choose **Yes** in the *Use Locale Format* and select the locale in the *Locale* field. Else, choose **No**.
10. Choose **Yes** in the *Enable Decimal Round Option* field if you want to round decimal values. Else, choose **No**.
11. Choose **User Id** or **Assignment ID** as the *Identity type*.  
Make sure that you select the same identity type as the one you chose when you exported the picklist.
12. Click **Import** and check the import status in the *Scheduled Job Manager* tool.

**Task overview:** [MDF Picklist Import \[page 119\]](#)

## Related Information

[Using ZIP Import with MDF Picklists \[page 121\]](#)

### 9.10.2.2 Using ZIP Import with MDF Picklists

Use ZIP import if you want to do full purge, change the picklist headers and values separately.

#### Context

If you want to do a full purge of your picklist, you must use the ZIP import option and import the picklist along with its picklist values.

#### Procedure

1. Prepare the ZIP file to be imported.
  - a. Change `purgeType=incremental` to `purgeType=fullpurge` in the `import.properties` file.
  - b. Add a `#` in front of the `suppressData=true` line in the `import.properties` file.
  - c. Compress `import.properties`, `import_sequence.csv`, `Picklist.csv`, and `Picklist-values.csv` into a zip file.
2. Select *Import Data* from the *Select the action to perform* dropdown list.
3. Choose the *ZIP File* tab.
4. Click the *Choose File* button and select the prepared zip file.
5. Click *Import* and check the import status in the *Scheduled Job Manager* tool.

**Task overview:** [MDF Picklist Import \[page 119\]](#)

## Related Information

[Using CSV Import with MDF Picklists \[page 120\]](#)

## 9.11 Making Picklist Value Fields Visible and Editable in Picklist Center

Make additional picklist value fields visible and/or editable in Picklist Center if the default migration settings don't meet your requirements.

### Prerequisites

You have the *Configure Object Definitions* permission, allowing you to manage MDF object definitions.

### Context

Some MDF picklist value fields are read-only or not visible in Picklist Center by default. You can make them visible and editable by updating the PicklistValue MDF object.

#### ❖ Example

The legacy picklist field `value` is used by SAP SuccessFactors Recruiting. During migration, it's mapped to the `optValue` field in MDF. If you're using SAP SuccessFactors Recruiting, you can make the `optValue` field editable in Picklist Center after picklist migration.

### Procedure

1. Go to ► *Admin Center* ► *Configure Object Definition* ▾.
2. Select *Object Definition* in the first search menu.
3. Type **Picklist Value** in the second search menu and select this object.
4. In the *Take Action* menu, click *Make Correction*.
5. In the *Fields* section, find the field you want to make editable in Picklist Center.
6. Click *Details* to see all of the properties of this field.
7. Set the *Visibility* property to *Read Only* or *Editable* as required.
8. Click *Done* to close the *Details* dialog and *Save* to save your changes.

### Results

You can now view and/or edit this field in Picklist Center.

## Note

The `optValue` field in the MDF object definition is called *Value* in Picklist Center because that was the original field name in legacy picklists.

## 9.12 Disabling Sync of Non-Unique External Code

Disable the automatic sync between the `nonUniqueExternalCode` and `externalCode` fields in MDF picklists.

### Prerequisites

You have access to Provisioning.

#### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

### Context

The `nonUniqueExternalCode` field in the `PicklistValue` MDF object contains the value of the `EXTERNAL_CODE` field from a migrated legacy picklist.

In many cases, we can't do a straight one-to-one mapping of external codes because the `EXTERNAL_CODE` in legacy picklists could be any value and it wasn't required, but the `externalCode` in MDF picklists is mandatory and unique. In such cases, we use to store the value of the legacy `EXTERNAL_CODE` so that applications that use it can ensure there's no change to the user experience.

By default, the value of the unique `externalCode` and the value of the `nonUniqueExternalCode` copied from legacy picklists are kept in sync in an MDF picklist. Any change you make to `externalCode` is automatically copied into `nonUniqueExternalCode` as well.

However, you can disable this behavior. If you used the `EXTERNAL_CODE` field in legacy picklists for a specific purpose that we couldn't anticipate and you do **not** want them to be synced automatically with the `externalCode` field in MDF picklists, you can choose to disable the sync.

### Procedure

1. Log in to Provisioning for the company.
2. Disable the option *Picklist Management: Automatically copy External Code to Non Unique External Code*.

## Results

The value of the `externalCode` field in MDF picklists is no longer copied automatically to the `nonUniqueExternalCode` field. Instead, they're treated as two distinct fields and you can use them for different purposes.

### 9.13 Field Mapping from Legacy to MDF Picklists

Legacy picklists have been migrated to MDF picklists. Learn how each field in legacy picklists was migrated to a corresponding field in MDF picklists.

Mapping of Legacy Picklists to MDF Picklists

| Legacy Picklist Field | Legacy Field Values  | MDF Object     | MDF Picklist Field Name | MDF Picklist Field Values  | Comments   |
|-----------------------|--|----------------|-------------------------|--|--|
| PICKLISTID            | <ul style="list-style-type: none"><li>• Must be unique</li><li>• Case-sensitive</li></ul>                                    | Picklist       | id (Code)               | <ul style="list-style-type: none"><li>• Must be unique.</li><li>• Case-insensitive</li></ul>               | In most cases, the <code>id</code> (or "Code") in MDF is the same as the legacy <code>PICKLISTID</code> . In the case of premigration conflicts, they can be different. For details, please refer to this Knowledge Base Article <a href="#">2770495</a> .   |
| PICKLISTID            | <ul style="list-style-type: none"><li>• Must be unique</li><li>• Case-sensitive</li></ul>                                    | Picklist       | legacyPickListId        | <ul style="list-style-type: none"><li>• Must be unique.</li><li>• Case-sensitive</li></ul>                 | The legacy <code>PICKLISTID</code> is always copied to the <code>legacyPickListId</code> field in MDF.   |
| STATUS                | Supports the following values: <ul style="list-style-type: none"><li>• ACTIVE</li><li>• OBSOLETE</li><li>• DELETED</li></ul> | Picklist Value | status                  | Supports the following values: <ul style="list-style-type: none"><li>• ACTIVE</li><li>• INACTIVE</li></ul> | Original legacy value is copied to <code>status</code> field in MDF. Values are mapped as follows: <ul style="list-style-type: none"><li>• ACTIVE in legacy maps to ACTIVE in MDF</li><li>• DELETED in legacy maps to INACTIVE in MDF</li><li>• OBSOLETE in legacy maps to INACTIVE in MDF</li></ul> |

| Legacy Picklist Field | Legacy Field Values  | MDF Object     | MDF Picklist Field Name | MDF Picklist Field Values   | Comments   |
|-----------------------|--|----------------|-------------------------|---|--|
| EXTERNAL_CODE         | <ul style="list-style-type: none"> <li>No constraints</li> <li>Not unique</li> <li>Can be null/empty.</li> </ul> | Picklist Value | externalCode            | <ul style="list-style-type: none"> <li>Must be unique</li> <li>Can't be null/empty</li> </ul> | In most cases, the externalCode in MDF is the same as the legacy EXTERNAL_CODE. Sometimes the externalCode value is changed during migration, following certain rules and conventions, to ensure that it's unique. |
| EXTERNAL_CODE         | <ul style="list-style-type: none"> <li>No constraints</li> <li>Not unique</li> <li>Can be null/empty.</li> </ul> | Picklist Value | nonUniqueExternalCode   | <ul style="list-style-type: none"> <li>Not unique</li> <li>Can be null/empty</li> </ul>       | Original legacy value is copied to nonUniqueExternalCode field in MDF.   |
| OPTIONID              | Unique primary key   | Picklist Value | N/A                     | N/A   | Original legacy value is copied to a mapping table but is still visible in the picklist object in MDF.   |
| VALUE                 | Present  | Picklist Value | optValue                | Not present before migration  | <p>Original legacy value is copied to optValue field in MDF.</p> <p>Before migration, the optValue isn't present in MDF. This field is created during migration to hold the legacy value.</p>                      |
| L_VALUE               | Present  | Picklist Value | lValue                  | Not present before migration  | <p>Original legacy value is copied to lValue field in MDF.</p> <p>Before migration, the lValue isn't present in MDF. This field is created during migration to hold the legacy value.</p>                          |



| Legacy Picklist Field | Legacy Field Values | MDF Object     | MDF Picklist Field Name | MDF Picklist Field Values    | Comments   |
|-----------------------|---------------------|----------------|-------------------------|------------------------------|--|
| R_VALUE               | Present             | Picklist Value | rValue                  | Not present before migration | Original legacy value is copied to rValue field in MDF.<br><br>Before migration, the rValue isn't present in MDF. This field is created during migration to hold the legacy value. |
| MINVAL                | Present             | Picklist Value | minVal                  | Not present before migration | Original legacy value is copied to minVal field in MDF.<br><br>Before migration, the minVal isn't present in MDF. This field is created during migration to hold the legacy value. |
| MAXVAL                | Present             | Picklist Value | maxVal                  | Not present before migration | Original legacy value is copied to maxVal field in MDF.<br><br>Before migration, the maxVal isn't present in MDF. This field is created during migration to hold the legacy value. |

# 10 Importing and Exporting Data

## 10.1 Exporting MDF Object Data

Exporting MDF object data allows you to perform instance migration, data analysis, mass editing and troubleshooting.

### Prerequisites

You have the ► [Admin Center Permissions](#) ► [Monitor Scheduled Jobs](#) ► permission to access [Scheduled Job Manager](#).

### Context

#### ⓘ Note

The export file only contains those object definitions for which corresponding features are enabled in your instance. While importing this export file in another instance, if for any of these object definitions, the corresponding feature isn't enabled, the import job fails, and error message is logged in logs file.

#### ⓘ Note

The export file will not include any attachment. To download the attachments of an MDF object, you need to first get the attachment IDs from the MDF OData API of the MDF object. Then use the OData API [Attachment](#) to export attachments. See [Exposing Custom MDF Objects to OData API \[page 209\]](#) and [Common Platform APIs: Attachment](#).

### Procedure

1. Go to ► [Admin Center](#) ► [Import and Export Data](#) ►.
2. Select [Export Data](#) from the [Select the action to perform](#) dropdown.
3. Select an object definition you want to export.
4. From the [Include Dependencies](#) dropdown, select **Yes** to include the dependencies else **No**.
5. To export the **Entity ID** and **Record ID**, select **Yes** from the [Include Immutable IDs](#) dropdown else select **No**.
6. To include inactive records, select **Yes** from the [Include Inactive Records](#) dropdown else select **No**.

7. To export the reference objects, select **No** from the *Exclude reference objects* dropdown else select **Yes**.
8. To export all the generic objects, select **Yes** from the *Select all data records* dropdown.

On selecting **No**, a *Select Object* dropdown appears and here you can specify the objects that you want to export.

9. From the *Key Preference* dropdown, select a key type.
10. To hide the external code, select **Yes** from the *Hide External Code* dropdown else select **No**.
11. Choose *User ID* or *Assignment ID* as the *Identity Type*.
12. Click *Export*.

An export job is triggered. You can choose *View Results* and navigate to *Scheduled Job Manager* to track the status of this job.

## Related Information

[Attachment](#)

[Common Platform APIs: Attachment](#)

## 10.2 Configuring RBP to Allow User to View Exports in Monitor Jobs

The download of exports from *Monitor Jobs* isn't read access logged even if there are any sensitive field included in the export. To view your own exports, Role-based Permission has to be set, otherwise any user can download the export of other users.

### Context

#### ⚠ Caution

We're planning to deprecate *Monitor Jobs*. As of the 1H 2022 release, you can monitor all MDF jobs in *Scheduled Job Manager*.

You can set up RBP to allow users to view only their own exports. By default, this Role-based permission isn't selected. To know whether the permission is selected or not, proceed as follows.

### Procedure

1. Go to [Admin Center](#) > [View User Permission](#).

2. Enter the *Username* or other required information and click *Search*.
3. Click *View Permission* from the search result. In this example, it's Admin.

The *Manage User Permission* dialog is displayed with the list of permissions for the user.

4. Search for *Allow users to view all the jobs. (By Disabling this option, users can view only their job status.)* to see if it's listed in the *Manage User Permission* dialog.

If yes, the searched admin can view all jobs on the Monitor Job page. To restrict Admin to view only the jobs submitted by Admin, you must disable this permission.

5. Go to [Admin Center](#) > [Manage Permission Roles](#) > [\[Permission Role\]](#) > [Permissions](#) > [Permission settings](#) > [Administrator Permission](#) > [Manage User](#).

All the permission related to Manage User is listed on the right pane.

6. Clear *Allow users to view all the jobs. (By Disabling this option, users can view only their job status.)* and click *Done*.

## 10.3 Exporting UI Configurations

You can export screens from one instance to another, provided the object and the localized data are the same.

### Prerequisites

- You have the [Admin Center Permissions](#) > [Monitor Scheduled Jobs](#) permission to access *Scheduled Job Manager*.
- You have the *Export Permission on Metadata Framework* permission to access *Import and Export Data*.
- You have the *Export* permission for the object you want to export.

### Procedure

1. Navigate to the [Admin Center](#) > [Import and Export Data](#).
2. Select *Export Data* from the *Select the action to perform* dropdown list.
3. Select the options from the dropdown lists of all mandatory fields to export the data.

| Field                 | Description                  |
|-----------------------|------------------------------|
| Select Generic Object | Select <i>ConfigUIMeta</i> . |

| Field                     | Description   |
|---------------------------|---|
| Include dependencies      | <p>Select one the following:</p> <ul style="list-style-type: none"> <li>• <a href="#">Yes</a></li> </ul> <p>This means that all CSV files corresponding to the parent object and related child objects are to be generated as a zip file.</p> <ul style="list-style-type: none"> <li>• <a href="#">No</a></li> </ul> <p>This means that only the CSV file corresponding to the object you select is to be generated as a zip file.</p> <div data-bbox="841 625 1435 850" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>Note</b></p> <p>An Import_Sequence.csv file is also generated in both options, containing information such as the name of the file and the sequence in which the CSV files are to be imported.</p> </div> |
| Include immutable IDs     | <p>Select one the following:</p> <ul style="list-style-type: none"> <li>• <a href="#">Yes</a></li> </ul> <p>This means that the entity ID and the record ID are exported.</p> <ul style="list-style-type: none"> <li>• <a href="#">No</a></li> </ul> <p>This means that the entity ID and the record ID aren't exported.</p>  |
| Include Inactive Records  | <p>Select <a href="#">Yes</a> or <a href="#">No</a> to decide whether you want to include inactive records.</p>   |
| Exclude reference objects | <p>Select one the following:</p> <ul style="list-style-type: none"> <li>• <a href="#">Yes</a></li> </ul> <p>This means that the reference objects aren't exported in the .zip file.</p> <ul style="list-style-type: none"> <li>• <a href="#">No</a></li> </ul> <p>This means that the reference objects are exported in the .zip file.</p>  |
| Select all data records   | <p>Choose <a href="#">Yes</a>.</p> <p>If you choose <a href="#">No</a>, an additional <a href="#">Select objects</a> field appears, where you can select the data records of your choice.</p>   |

| Field          | Description   |
|----------------|---|
| Key Preference | <ul style="list-style-type: none"> <li>If you only want business key in the data, choose <i>Business Key</i>, and also choose <i>Yes</i> in the <i>Hide External Code</i> field.</li> <li>If you don't want business key in the data, choose <i>External Code</i>.</li> <li>If you want both business key and external code, choose <i>Business Key</i>, and also choose <i>No</i> in the <i>Hide External Code</i> field.</li> </ul> |
| Identity type  | Choose <i>User ID</i> or <i>Assginment ID</i> as the identity type.   |

## Results

You can choose [View Results](#) and navigate to [Scheduled Job Manager](#) to check the result.

## 10.4 Importing MDF Data

This section explains how to import MDF data.

You can easily import MDF objects and related object hierarchy. You can import and export data to and from your client's MDF instance. MDF imports leverage an in-memory processing approach instead of a staging table, which improves performance.

Importing data to an MDF instance is a two-step process. First, you download the correct template for the data, and enter the data that you want to import. The template contains the structure of the object because it's set up in MDF object definitions. Second, you upload the data for that object. Both these operations use the same form that is launched.

You can choose [View Results](#) and navigate to [Scheduled Job Manager](#) to check the progress of import jobs.

### Note

All decimals are rounded to the precision defined in the object definition as of the 2H 2022 release.

### Note

As of Q3 2017 release, the original and the cloned objects have different references of attachment. So, deleting the original object doesn't delete the attachments referenced by cloned object.

### Caution

During the incremental import of any MDF object, records that have pending workflow approval are imported and "Pending data exist!" error is returned. To address this issue, user must either complete or cancel workflow

by either Approving or rejecting and try importing again. Alternately you can remove the rejected records from import file and complete the import for rest of the records in the file.

#### [Importing Data Purging Options: Incremental or Full Purge \[page 132\]](#)

Understand how different purging options work to import data.

#### [Configuring Batch Size for Importing MDF Entities \[page 137\]](#)

Configure the batch size for importing MDF entities in the UI to avoid transaction timeout exception or issues in certain scenarios, such as importing objects with a large number of rules.

#### [Suppress Duplicate Inserts \[page 137\]](#)

In the MDF import, you can suppress duplicate inserts and unchanged updates, in incremental mode. This can help eliminate millions of redundant user data while improving performance.

#### [Using Delimit Functionality for MDF Entities \[page 139\]](#)

In the MDF data import, DELIMIT functionality is supported for all entities (effective-dated and non-effective-dated) that have multiple records for the same user or person, for example, phone information, social account information, e-mail information, address information, and so on.

#### [Valid Values for Enum and Boolean Data Type Fields in Import Template \[page 145\]](#)

Valid values for Enum data type fields are available along with Boolean data type fields in the MDF import template. For Enum data type fields, you can specify exact values.

#### [Validating MDF Data \[page 146\]](#)

You can check the import file for errors and inconsistencies before importing the MDF data. This helps reduce the risk of failures during the actual import process.

#### [Deleting Records for MDF Objects Through Import \[page 147\]](#)

You can now delete all the records for an MDF object such as instances and entities using *Delete* operator in imports.

#### [Adding Attachments to MDF Objects During Data Import \[page 149\]](#)

You can add attachments to MDF objects that have a field of data type Attachments during data import.

## 10.4.1 Importing Data Purging Options: Incremental or Full Purge

Understand how different purging options work to import data.

Incremental purge only add new records and update values of old records. Full purge deletes all existing records and create new records as specified in the upload file.

None effective dated objects only keep one record without timeslice information. The results are the same using either purging option when importing data for none effective dated objects.

### Example Object and Upload File

To demonstrate the difference between the two purging options on effective dated objects, we use `Position` as an example. Entities and their records are listed in the following table.

#### Example Entities of Effective Dated Object

| Entity Code | Effective Start Date | Record Name              |
|-------------|----------------------|--------------------------|
| Position 1  | 1990-01-01           | Java Developer           |
| Position 1  | 2000-01-01           | Software Developer       |
| Position 2  | 1990-01-01           | C++ Developer            |
| Position 2  | 2000-01-01           | C# Developer             |
| Position 2  | 2010-01-01           | Senior Software Engineer |
| Position A  | 1990-01-01           | Sales                    |

There are three entities of `Position`: Position 1, Position 2, and Position A.

- Position 1 has two records with effective start dates in 1990 and 2000.
- Position 2 has three records with effective start dates in 1990, 2000, and 2010.
- Position A has only one record with an effective start date in 1990.

We prepared an upload file with entities and records listed in the following table.

#### Example Upload File

| Entity Code | Effective Start Date | Record Name            |
|-------------|----------------------|------------------------|
| Position 1  | 2010-01-01           | Software Engineer - L1 |
| Position 2  | 2000-01-01           | Software Engineer - L2 |
| Position 3  | 2010-01-01           | Software Engineer - L3 |

## Incremental Purge

Importing data by incremental purge works at the record level. If we import the example file to the system by incremental purge, we get the following result.

- If a record in the upload file finds no record in the system with the same entity code and effective start date, this record is added as a new record in the system. See emphasized entries of Position 1 and 3.
- If a record in the upload file matches an existing record meaning that they have the same entity code and effective start date, the old record is replaced by the record in the upload file. See the emphasized entry of Position 2.
- Other existing records remain unchanged in the system.

#### Result of Incremental Purge

| Entity Code       | Effective Start Date | Record Name                   |
|-------------------|----------------------|-------------------------------|
| Position 1        | 1990-01-01           | Java Developer                |
| Position 1        | 2000-01-01           | Software Developer            |
| <b>Position 1</b> | <b>2010-01-01</b>    | <b>Software Engineer - L1</b> |
| Position 2        | 1990-01-01           | C++ Developer                 |
| <b>Position 2</b> | <b>2000-01-01</b>    | <b>Software Engineer - L2</b> |



| Entity Code | Effective Start Date | Record Name            |
|-------------|----------------------|------------------------|
| Position 3  | 2010-01-01           | Software Engineer - L3 |
| Position A  | 1990-01-01           | Sales                  |

## Full Purge

Importing data by full purge works at the entity level. If we import the example file to the system by full purge, we get the following result.

- Existing records of the same entities are deleted from the system and replaced by entries in the upload file.
- Existing records that aren't specified in the upload file remain unchanged in the system. See the entry of Position A.

Result of Full Purge

| Entity Code | Effective Start Date | Record Name            |
|-------------|----------------------|------------------------|
| Position 1  | 2010-01-01           | Software Engineer - L1 |
| Position 2  | 2000-01-01           | Software Engineer - L2 |
| Position 3  | 2010-01-01           | Software Engineer - L3 |
| Position A  | 1990-01-01           | Sales                  |

Parent topic: [Importing MDF Data \[page 131\]](#)

## Related Information

[Configuring Batch Size for Importing MDF Entities \[page 137\]](#)

[Suppress Duplicate Inserts \[page 137\]](#)

[Using Delimit Functionality for MDF Entities \[page 139\]](#)

[Valid Values for Enum and Boolean Data Type Fields in Import Template \[page 145\]](#)

[Validating MDF Data \[page 146\]](#)

[Deleting Records for MDF Objects Through Import \[page 147\]](#)

[Adding Attachments to MDF Objects During Data Import \[page 149\]](#)

### 10.4.1.1 Importing Data Using CSV Files for Parent and Child Objects

Understand the impact of incremental and full purge on parent and child objects.

Without a good understanding of the purge behavior and the data model, you risk deleting records that are referenced by other objects and causing data corruption.

## Importing Parent Entities Only

When you select the parent object as the target object and only include entities of the parent object in your upload file, the full purge option deletes all existing records and associated data of entities specified in the file and creates new records as specified in the file, which isn't desired in most cases.

### Note

- The full purge option is turned off for importing picklists.
- We recommend incremental purge when you import parent objects only.

When you select the parent object as the target object and only include entities of the parent object in your upload file, the incremental purge option can achieve the following results:

- You update existing records of the parent entity.
- You add an additional record to the parent entity. If there's any existing record of that parent entity, all associations in that record are copied over to the new record.

### Note

If you don't want those copied-over associations, use the ZIP file import with both parent and child objects in the upload file.

## Importing Child Entities Only

When you select the child object as the target object and only include records of the association in your upload file, the full purge option replaces all existing association records with what's in the upload file.

When you select the child object as the target object and only include records of the association in your upload file, the incremental purge option adds new child records or updates existing child records.

### Note

Make sure that the parent entity or parent records listed in the upload file exist in the system. Otherwise, the import could fail because of missing values for mandatory fields.

Even when you successfully create a new parent record by importing child entities only, all associations of the previous record are copied over to the new record, which isn't desirable in most cases.

## Example

We have the following records in the system:

Example Parent and Child Records

| cust_parent Code | cust_parent Effective Start Date | cust_child Code | cust_child Name |
|------------------|----------------------------------|-----------------|-----------------|
| P1               | 1990-01-01                       | C1              | V1              |
| P1               | 1990-01-01                       | C2              | V2              |
| P1               | 2000-01-01                       | C1              | V1              |
| P1               | 2000-01-01                       | C2              | V2              |
| P2               | 1990-01-01                       | C1              | V1              |

We have an upload file with only association records as follows:

Example Upload File

| cust_parent Code | cust_parent Effective Start Date | cust_child Code | cust_child Name |
|------------------|----------------------------------|-----------------|-----------------|
| P1               | 2000-01-01                       | C1              | V1_UPDATE       |
| P1               | 2000-01-01                       | C3_NEW          | V3_NEW          |

If we choose the incremental purge option, we get the following result:

Result of Incremental Purge

| cust_parent Code | cust_parent Effective Start Date | cust_child Code | cust_child Name  |
|------------------|----------------------------------|-----------------|------------------|
| P1               | 1990-01-01                       | C1              | V1               |
| P1               | 1990-01-01                       | C2              | V2               |
| <b>P1</b>        | <b>2000-01-01</b>                | <b>C1</b>       | <b>V1_UPDATE</b> |
| P1               | 2000-01-01                       | C2              | V2               |
| <b>P1</b>        | <b>2000-01-01</b>                | <b>C3_NEW</b>   | <b>V3_NEW</b>    |
| P2               | 1990-01-01                       | C1              | V1               |

If we choose the full purge option, we get the following result:

Result of Full Purge

| cust_parent Code | cust_parent Effective Start Date | cust_child Code | cust_child Name  |
|------------------|----------------------------------|-----------------|------------------|
| P1               | 1990-01-01                       | C1              | V1               |
| P1               | 1990-01-01                       | C2              | V2               |
| <b>P1</b>        | <b>2000-01-01</b>                | <b>C1</b>       | <b>V1_UPDATE</b> |
| <b>P1</b>        | <b>2000-01-01</b>                | <b>C3_NEW</b>   | <b>V3_NEW</b>    |
| P2               | 1990-01-01                       | C1              | V1               |

## 10.4.2 Configuring Batch Size for Importing MDF Entities

Configure the batch size for importing MDF entities in the UI to avoid transaction timeout exception or issues in certain scenarios, such as importing objects with a large number of rules.

### Procedure

1. Go to the ► [Admin Center](#) ► [Import and Export Data](#) ▾.
2. Select [Import Config](#) as the action to perform.
3. Select the generic object you want to import later.
4. Enter the batch size.

The default is 500, but it can be from 20 through 2000.

**Task overview:** [Importing MDF Data \[page 131\]](#)

### Related Information

[Importing Data Purging Options: Incremental or Full Purge \[page 132\]](#)

[Suppress Duplicate Inserts \[page 137\]](#)

[Using Delimit Functionality for MDF Entities \[page 139\]](#)

[Valid Values for Enum and Boolean Data Type Fields in Import Template \[page 145\]](#)

[Validating MDF Data \[page 146\]](#)

[Deleting Records for MDF Objects Through Import \[page 147\]](#)

[Adding Attachments to MDF Objects During Data Import \[page 149\]](#)

## 10.4.3 Suppress Duplicate Inserts

In the MDF import, you can suppress duplicate inserts and unchanged updates, in incremental mode. This can help eliminate millions of redundant user data while improving performance.

For example, in an effective-dated entity import, if the newly inserted time record is identical to its former dated record (where both field values and composite or valid-when relationships), no new record is generated by the import.

## Import and Export Data

\* Select the action to perform 

## Import Data

 CSV File
  ZIP File
  Success Store

|   |   |
|---|---|
| * Select Generic Object                   | <input type="text" value="Business Unit"/>                        |
| * File                                    | desktop\BusinessUnit.csv <input type="button" value="Browse..."/> |
| File Encoding                             | <input type="text" value="Western European (Windows/ISO)"/>       |
| Purge Type                                | <input type="text" value="Incremental Load"/>                     |
| Suppress Redundant Date-Effective Records | <input type="text" value="Yes"/>                                  |
| Key Preference                            | <input type="text" value="Business Key"/>                         |
| Use Locale Format                         | <input type="text" value="No"/>                                   |
| Enable Decimal Round Option               | <input type="text" value="Yes"/>                                  |
| Date Format                               | MM/DD/YYYY  |

**Note**

For MDF imports, to avoid accumulation of redundant data by duplicate inserts and unchanged updates, by default the *Suppress Redundant Date-Effective Records* option is enabled. This eliminates redundant date-effective records and improves performance. You have the option to opt-out of *Suppress Redundant Date-Effective Records* by selecting *No* in the *Import and Export Data* page, if that is decided for the concerned customer or instance.

For effective-dated entity imports (for example, position import), if the newly inserted time record is identical to its former dated record (both field values and composite or valid-when relationships), no new record are generated with import. This special handling feature is activated by an own mode flag in the import. For example, there is a position in the database whose effectiveStartDate is 1-1-2015. Now try to insert a position on 2-1-2015. If every field in the csv file has identical value with that position of 1-1-2015 and its composite relationship is same with the position 1-1-2015, this position 2-1-2015 is not generated with the import. So nothing will change once the import is done.

**Note**

For successful zip import, set **suppressData** to **false** or delete **suppressData=true** from **import.properties** file. If you set **suppressData** to **true**, an error message is displayed.

For Full purge, by default **suppressData** is set to **false**. For incremental, by default **suppressData** is set to **true**.

Parent topic: [Importing MDF Data \[page 131\]](#)

## Related Information

[Importing Data Purging Options: Incremental or Full Purge \[page 132\]](#)

[Configuring Batch Size for Importing MDF Entities \[page 137\]](#)

[Using Delimit Functionality for MDF Entities \[page 139\]](#)

[Valid Values for Enum and Boolean Data Type Fields in Import Template \[page 145\]](#)

[Validating MDF Data \[page 146\]](#)

[Deleting Records for MDF Objects Through Import \[page 147\]](#)

[Adding Attachments to MDF Objects During Data Import \[page 149\]](#)

## 10.4.4 Using Delimit Functionality for MDF Entities

In the MDF data import, DELIMIT functionality is supported for all entities (effective-dated and non-effective-dated) that have multiple records for the same user or person, for example, phone information, social account information, e-mail information, address information, and so on.

### Context

In the *Operations* column in the import template, if a row has the value DELIMIT, that record is deleted in the case of non-effective-dated entities, and it is updated with the effective end date in the case of effective-dated entities.

If data is imported with no values for an association in an incremental import, delimited records will be created and association data will also be cleared and not retained.

For example, in the corporate data model, a location is associated with the cost center as ONE\_TO\_MANY relationship. You import the cost center value CC1 in full purge mode with the associated locations Loc1, Loc2, and Loc3 on 15-Jan-2014. You then import a new record for CC1 on 15-Jan-2015 with no location values. The import file for the location is created with blanks for the delimited record.

### Procedure

1. Go to the *Admin Center*.
2. In the *Tools search* field, type *Import Data*.
3. In the *Import and Export Data* screen, from the *Select action to perform* dropdown list, choose *Download Template*.
4. In the form, select the following:

Back to: Admin Center

### Import and Export Data

\* Select the action to perform

---

#### Download Template

\* Select Generic Object

\* Include dependencies

\* Include Immutable IDs

\* Exclude reference objects

| Field                     | Description   |
|---------------------------|---|
| Select Generic Object     | Select the required generic object.   |
| Include dependencies      | <p>Select one the following:</p> <ul style="list-style-type: none"> <li>• <a href="#">Yes</a></li> </ul> <p>For the package to be generated as a zip file of all CSV files corresponding to the parent object and related child objects.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>An Import_Sequence.csv file is also generated, containing information such as the name of the file and the sequence in which the CSV files are to be imported.</p> </div> <ul style="list-style-type: none"> <li>• <a href="#">No</a></li> </ul> |
| Include immutable IDs     | <p>Select one the following:</p> <ul style="list-style-type: none"> <li>• <a href="#">Yes</a></li> </ul> <p>This means that the entity ID and the record ID are imported.</p> <ul style="list-style-type: none"> <li>• <a href="#">No</a></li> </ul> <p>This means that the entity ID and the record ID are not imported.</p>   |
| Exclude reference objects | <p>Select one the following:</p> <ul style="list-style-type: none"> <li>• <a href="#">Yes</a></li> </ul> <p>This means that the reference objects are not imported in the .zip file.</p> <ul style="list-style-type: none"> <li>• <a href="#">No</a></li> </ul> <p>This means that the reference objects are imported in the .zip file.</p>   |

5. Click [Download](#).
6. Open the CSV file that is downloaded and enter the data that you want to import for the object.

If you set default values for field in the object definition, you can also set MDF fields to that default value during import. Simply enter the placeholder value "&&DEFAULT&&" in the CSV import file and the default value will be displayed. You can even have placeholders for auto-generated externalCode and transactionSequence fields.

7. Save your data.

MDF supports the NO\_OVERWRITE function for import of any MDF object. For any field value marked as NO\_OVERWRITE, the existing value is either retained or carried over and updated by the system with the values provided by the user. This is applicable for both effective-dated and non-effective-dated entities. Earlier, during MDF import, this was possible only if the user provided the values while updating a record.

**Note**

You must provide values for the fields that you want to change or those that are part of a business key to uniquely identify the record. This is required for ECv2 import parity.

8. On the *Import and Export Data* page, select *Import Data* as the action to perform.

9. Select one of the following options:

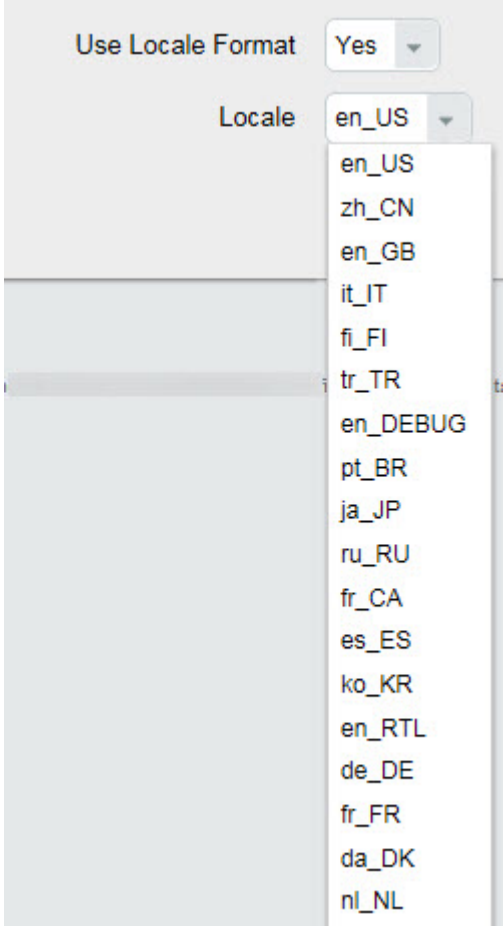
a. **CSV File**

The following *Import Data* form appears.

| Field                 | Description   |
|-----------------------|---|
| Select Generic Object | Select the required generic object.                 |
| File                  | Browse and choose the file that you want to upload. |
| File Encoding         | Choose the correct file encoding for your data.     |



| Field          | Description  |
|----------------|--|
| Purge Type     | <p>Choose the required purge type:</p> <ul style="list-style-type: none"> <li>• <i>Full Purge</i>. Use this option to delete all existing data within the application. You can only retain records available in your import file.</li> </ul> <div data-bbox="911 443 1425 701" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>Note</b></p> <p>If an object has a composite association, do not use this option in addition to the CSV import operation. By doing so, you will delete all child records associated with a parent record in the CSV file.</p> </div> <ul style="list-style-type: none"> <li>• <i>Incremental Load</i>. Use this option to add or update records in the CSV file without deleting other effective-dated records.</li> </ul> <p>The records for the external code in the import file are either inserted or updated. The records are updated when an existing record is found with the same external code, start date, and transaction sequence number (for MCPD). All global information records for the user and start date are also deleted. All other records are unchanged.</p> <p>For more information, see the <a href="#">Implementing Employee Central Core</a> from the SAP Help Portal.</p> |
| Key Preference | <p>The key preference option is added to ensure backward compatibility when a new record is added the same externalCode and a different business key from an existing record.</p> <p>Select on of the following:</p> <ul style="list-style-type: none"> <li>• External Code<br/>Choose this to update the business key.</li> <li>• Business Key</li> </ul> <div data-bbox="911 1455 1425 1644" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>Note</b></p> <p>A business key for an MDF object is a set of fields of the MDF object that can be used as a unique key.</p> </div> <p>The import will fail since the new business key is considered a new record - and for this record, there is already an externalCode.</p>  |

| Field             | Description  |
|-------------------|--|
| Use Locale Format | <p>Select one of the following:</p> <ul style="list-style-type: none"> <li>Yes           <p>Choose this to set dates and numbers to specific locale formats. Another field appears for you to choose the locale.</p>  </li> <li>No</li> </ul> |
| Date Format       | MM/DD/YYYY   |

To validate the data before importing it, click [Validate](#); otherwise, click [Import](#).

b. **ZIP File**

The following [Import Data](#) form appears.

Back to: Admin Center

### Import and Export Data

\* Select the action to perform

#### Import Data

CSV File
  ZIP File
  Success Store

\* File

Date Format: MMDDYYYY

Choose the required file. To validate the data before importing it, click [Validate](#); otherwise, click [Import](#).

#### Note

To use MDF zip import, ensure that the zip only includes files whose names do not contain non-UTF-8 characters. Importing data is not supported when the filename of the zipped files contain non-UTF-8 characters.

#### c. Success Store

The following [Import Data](#) form appears.

Import and Export Data

\* Select the action to perform

#### Import Data

CSV File
  ZIP File
  Success Store

|                                  | Package Name              | Type                       | Package Description                   |
|----------------------------------|---------------------------|----------------------------|---------------------------------------|
| <input type="radio"/>            | 111                       | LarryTest_MultiAssociation | 111                                   |
| <input checked="" type="radio"/> | Holiday Calendar          | Time Off                   | Holiday Calendar                      |
| <input type="radio"/>            | Holiday_AB                |                            | HOLIDAY                               |
| <input type="radio"/>            | Larry Test 2              | MDF                        | Larry Test 2                          |
| <input type="radio"/>            | Larry Test 3              | Package Import             | Multiple Level Testing                |
| <input type="radio"/>            | Larry Test1               | aaa                        | Larry Test1                           |
| <input type="radio"/>            | MDF Sample Objects        | Timeoff                    | MDF Sample Objects description        |
| <input type="radio"/>            | Second MDF Sample Objects | Position Management        | Second MDF Sample Objects description |
| <input type="radio"/>            | Test                      |                            | test                                  |

Select the required packages and click [Import](#).

#### Note

On import, the system will validate the effectiveStartDate to check if the source is in range with target effectiveStartDate and effectiveEndDate for the following:

- Generic object - field
- Generic object - validwhen
- Foundation object - field
- Picklist - field

If the source is not in range with what is expected, an error message appears in Job Response file.

Check out the result in [Scheduled Job Manager](#).

**Task overview:** [Importing MDF Data \[page 131\]](#)

## Related Information

[Importing Data Purging Options: Incremental or Full Purge \[page 132\]](#)

[Configuring Batch Size for Importing MDF Entities \[page 137\]](#)

[Suppress Duplicate Inserts \[page 137\]](#)

[Valid Values for Enum and Boolean Data Type Fields in Import Template \[page 145\]](#)

[Validating MDF Data \[page 146\]](#)

[Deleting Records for MDF Objects Through Import \[page 147\]](#)

[Adding Attachments to MDF Objects During Data Import \[page 149\]](#)

## 10.4.5 Valid Values for Enum and Boolean Data Type Fields in Import Template

Valid values for Enum data type fields are available along with Boolean data type fields in the MDF import template. For Enum data type fields, you can specify exact values.

The following figure shows valid values for Enum columns.

| [OPERATOR]                         | externalCode | externalName | effectiveStartDate |
|------------------------------------|--------------|--------------|--------------------|
| Supported operators: Delimit Clear | externalCode | externalName | effectiveStartDate |
|                                    |              |              |                    |

**Parent topic:** [Importing MDF Data \[page 131\]](#)

## Related Information

[Importing Data Purging Options: Incremental or Full Purge \[page 132\]](#)

[Configuring Batch Size for Importing MDF Entities \[page 137\]](#)

[Suppress Duplicate Inserts \[page 137\]](#)

[Using Delimit Functionality for MDF Entities \[page 139\]](#)

[Validating MDF Data \[page 146\]](#)

[Deleting Records for MDF Objects Through Import \[page 147\]](#)

[Adding Attachments to MDF Objects During Data Import \[page 149\]](#)

## 10.4.6 Validating MDF Data

You can check the import file for errors and inconsistencies before importing the MDF data. This helps reduce the risk of failures during the actual import process.

### Context

The *Validate* option performs basic checks on your import file, such as file format, column count, date format, and duplicate records. However, it doesn't catch all potential import issues. You may still encounter errors during the actual import process, even if the validation passes. When importing multiple objects using a Zip file, validation may show errors for referenced data that exists only within the import file and not yet in the database.

For example, if you're importing *Position* and *Job Code* together, the validation action flags an error if a Position references a Job Code that is only in the import file (and not yet in the system). This is because the system can't find the Job Code reference in the database during validation. However, the actual import may succeed, as the Job Code is imported before the Position.

### Procedure

1. Go to ► [Admin Center](#) ► [Import and Export Data](#) ►.
2. Select *Import Data* from the *Select the action to perform* dropdown.
3. You can choose *Validate* to check if there's any errors in your import file.

#### Note

During an import, the *Validate* option checks against the data within the import file. So, a successful validation doesn't guarantee a successful import if there are cross-object dependencies within the zip file.

**Task overview:** [Importing MDF Data \[page 131\]](#)

### Related Information

[Importing Data Purging Options: Incremental or Full Purge \[page 132\]](#)

[Configuring Batch Size for Importing MDF Entities \[page 137\]](#)

[Suppress Duplicate Inserts \[page 137\]](#)

[Using Delimit Functionality for MDF Entities \[page 139\]](#)

[Valid Values for Enum and Boolean Data Type Fields in Import Template \[page 145\]](#)

[Deleting Records for MDF Objects Through Import \[page 147\]](#)

[Adding Attachments to MDF Objects During Data Import \[page 149\]](#)

## 10.4.7 Deleting Records for MDF Objects Through Import

You can now delete all the records for an MDF object such as instances and entities using *Delete* operator in imports.

### Context

To delete the root instance along with the associations through MDF Import, proceed as follows:

### Procedure

1. Export the instance data. For example, `DeleteRecord.csv`.

| [OPERATOR]                         | externalCode | externalName | effectiveStartDate |
|------------------------------------|--------------|--------------|--------------------|
| Supported operators: Delimit Clear | externalCode | externalName | effectiveStartDate |
|                                    | root1        |              | 1/1/2017           |
|                                    | root1        |              | 2/2/2017           |
|                                    | root2        |              | 1/1/2017           |

Root Template

| [OPERATOR]                         | externalCode           | effectiveStartDate           | cust_comp.externalCode | cust_comp.externalName |
|------------------------------------|------------------------|------------------------------|------------------------|------------------------|
| Supported operators: Delimit Clear | cust_Name.externalCode | cust_Name.effectiveStartDate | externalCode           | externalName           |
|                                    | root1                  | 1/1/2017                     | compA                  | nameA                  |
|                                    | root1                  | 1/1/2017                     | compB                  | nameB                  |
|                                    | root1                  | 2/2/2017                     | compC                  | nameC                  |
|                                    | root1                  | 2/2/2017                     | compD                  | nameD                  |
|                                    | root1                  | 1/1/2017                     | comp1                  | name1                  |
|                                    | root2                  | 1/1/2017                     | comp2                  | name2                  |

Composite Association

2. Add *Delete* operator in the root template for the record that you want to delete from the instance.

| [OPERATOR]                         | externalCode | externalName | effectiveStartDate |
|------------------------------------|--------------|--------------|--------------------|
| Supported operators: Delimit Clear | externalCode | externalName | effectiveStartDate |
| Delete                             | root1        |              | 2/2/2017           |

Delete Operator in the Root Template

3. Import the `DeleteRecord.csv` file.

After the file is imported successfully, the record for which the *Delete* operator was added is deleted.

| [OPERATOR]                         | externalCode | externalName | effectiveStartDate |
|------------------------------------|--------------|--------------|--------------------|
| Supported operators: Delimit Clear | externalCode | externalName | effectiveStartDate |
|                                    | root1        |              | 1/1/2017           |
|                                    | root2        |              | 1/1/2017           |

Root Instance Data After Import (Deletion)

| [OPERATOR]                         | externalCode           | effectiveStartDate           | cust_comp.externalCode | cust_comp.externalName |
|------------------------------------|------------------------|------------------------------|------------------------|------------------------|
| Supported operators: Delimit Clear | cust_Name.externalCode | cust_Name.effectiveStartDate | externalCode           | externalName           |
|                                    | root1                  | 1/1/2017                     | compA                  | nameA                  |
|                                    | root1                  | 1/1/2017                     | compB                  | nameB                  |
|                                    | root1                  | 1/1/2017                     | comp1                  | name1                  |
|                                    | root2                  | 1/1/2017                     | comp2                  | name2                  |

Composite Association Data After Import (Deletion)

### Note

- The *Delete* operator is only allowed for Root template. Adding Delete operator in the root template, deletes the associated composite child.
- If *Delete* operator is added for Root template, *Correct*, *Delimit*, and *Clear* operators are not supported for that record as this causes unsuccessful import.
- If *Delete* operator is added for Root template, *Correct* and *Insert* operators are not supported on the same effective dated records of different time slices.
- If *Delete* operator is added for a root template which has **valid when association** then **valid when instance** is not deleted. You have to delete them separately.
- If you want to perform batch deletion for root template, you have to add *Delete* operator for all the records that you want to delete and then perform import.

| [OPERATOR]                         | externalCode | externalName | effectiveStartDate |
|------------------------------------|--------------|--------------|--------------------|
| Supported operators: Delimit Clear | externalCode | externalName | effectiveStartDate |
| Delete                             | root1        |              | 1/1/2017           |
| Delete                             | root2        |              | 1/2/2017           |
| Delete                             | root3        |              | 1/3/2017           |
| Delete                             | root4        |              | 1/4/2017           |
| Delete                             | root5        |              | 1/5/2017           |
| Delete                             | root6        |              | 1/6/2017           |
| Delete                             | root7        |              | 1/7/2017           |
| Delete                             | root8        |              | 1/8/2017           |
| Delete                             | root9        |              | 1/9/2017           |
| Delete                             | root10       |              | 1/10/2017          |

Batch Deletion

Task overview: [Importing MDF Data \[page 131\]](#)

## Related Information

[Importing Data Purging Options: Incremental or Full Purge \[page 132\]](#)

[Configuring Batch Size for Importing MDF Entities \[page 137\]](#)

[Suppress Duplicate Inserts \[page 137\]](#)

[Using Delimit Functionality for MDF Entities \[page 139\]](#)

[Valid Values for Enum and Boolean Data Type Fields in Import Template \[page 145\]](#)

[Validating MDF Data \[page 146\]](#)

[Adding Attachments to MDF Objects During Data Import \[page 149\]](#)

## 10.4.8 Adding Attachments to MDF Objects During Data Import

You can add attachments to MDF objects that have a field of data type Attachments during data import.

### Context

Before you import attachments, read the following notes:

- You're only allowed to link one attachment file to one record. If the same content is required by multiple records, you must create multiple copies of same attachment file with unique names and link them individually to each record. However, different effective-dated records of the same object can still share the same attachment.
- MDF data import doesn't trigger workflows.

### Procedure

1. Go to ► [Admin Center](#) ► [Import and Export Data](#) ►.
2. Select [Download Template](#) from the [Select the action to perform](#) dropdown.
3. Enter the required information for the MDF Object that has [Attachments](#) type field.
4. Click [Download](#).

A ZIP file gets downloaded to the default download location.

5. Extract the ZIP file. Go to the extracted folder, create a new folder inside and name it **attachments**.

#### ⚠ Caution

The new folder only works if it's named exactly as **attachments**. Incorrect folder name prevents the files from uploading.

6. Copy all the attachments to this folder that is required to be added on the MDF Object.
7. Open the .csv file for the MDF Object and enter the corresponding file name along with the correct extension in the **Attachments** field as shown.

|   | A                 | B            | C            | D                    | E                 | F                |
|---|-------------------|--------------|--------------|----------------------|-------------------|------------------|
| 1 | [OPERATOR]        | externalCode | externalName | cust_division.extern | mdfSystemStatus   | cust_attachments |
| 2 | Supported operato | externalCode | externalName | Division.Code        | mdfSystemStatus() | attachments      |
| 3 |                   | cgrant1      | Carla Grant  | HC                   | A                 | &&NO_OVERWRITE&& |
| 4 |                   | admin1       | admin admin  |                      | A                 | Quiz.docx        |
| 5 |                   |              |              |                      |                   |                  |

Attachments for an MDF Object in the CSV File

8. Save the changes and create a ZIP file of all the files and the attachment folder.

#### ⚠ Caution

Always use standard ZIP format. The ZIP file must be created within the extracted folder by selecting all files and folders within it. If you compress the parent folder, you'll get an error during upload.



9. Go to [Admin Center](#) > [Import and Export Data](#).
10. Select [Import Data](#) from the *Select the action to perform* dropdown.
11. Optionally, you can choose [Validate](#) to check if there's any errors in the upload content.

#### Note

If there's a validate rule about the attachment, the validation process might fail. To prevent such failures, configure the rule to be skipped during import. For more information, see the Related Information section.

12. In the [Import Data](#) section, select [ZIP File](#) option.
13. Browse for the updated ZIP file and click [Import](#).

The attachment gets added to the MDF Object.

**Task overview:** [Importing MDF Data \[page 131\]](#)

## Related Information

[Importing Data Purging Options: Incremental or Full Purge \[page 132\]](#)

[Configuring Batch Size for Importing MDF Entities \[page 137\]](#)

[Suppress Duplicate Inserts \[page 137\]](#)

[Using Delimit Functionality for MDF Entities \[page 139\]](#)

[Valid Values for Enum and Boolean Data Type Fields in Import Template \[page 145\]](#)

[Validating MDF Data \[page 146\]](#)

[Deleting Records for MDF Objects Through Import \[page 147\]](#)

[Excluding Rule Execution during Import \[page 183\]](#)

## 10.5 Importing Associated Objects

You can download a template for associated objects and fill in data along with the parent object externalCode.

You can also import all associated objects in a single template, but you will need to download the template for all associated objects and manually add all columns of the associated objects along with the columns of the parent object to prepare a single template.

There is a valid-when association between the current object and the referred target, where “current” is the host GO entity and “referred target” is the association destination target entity.

If both the current object and referred target allow effective dating (basic or MCPD), the effective\_start\_date of the referred target must not be later than the effective\_start\_date of the current object (that is, it must be earlier or equal). Any row that violates this constraint in the CSV file will not be imported. In the above example, if the effective\_start\_date of the parent position is later than that of the current position, the position will not be imported successfully and an error message will be returned in the import status file.

## Example

In the *Position* object, there is a valid-when association as parentPosition. The current position is the host/current object and the referred target is the parent position value.

## 10.6 Rule Integration with MDF Import

MDF allows end user to add business logic to an object using rules. Like UI, rules can also be triggered using imports. This is done by using the MDF Import tool.

MDF Import tool allows you to add rules for both parent MDF as well as composite child entities.

You can use the Validate option in the Import and Export Data tool to validate the MDF data before importing a file. The Validate option, triggers the validate rules associated with the MDF objects you want to import. If there are any errors in the Import file, due to validate rule you're notified about them. This lets you catch and rectify any data issues before importing it.

Not all rule types are supported in MDF Import tool, supported rule types are:

- **Initialize Rules**  
Initialization rules are triggered in the initialization phase of an object creation.
- **Validate Rules**  
Validate rules are triggered after a change to an object is submitted but before the change is saved.
- **Save Rules**  
Save rules are triggered when an object is being saved.

### 10.6.1 Initialize Rules

Initialization rules ("initializeRules") are triggered in the initialization phase of object creation (an "onInit" event). You can use these rules to initialize fields to default values. For example, initialization rules can be used to autofill employee ID field with the login user ID. Note that the Initialization rules are not supported when setting the wfConfig attribute to trigger workflows.

Once the initialize rule is added to the object definition, you can fill the non-existing external code and all other required fields in import file.

#### ⓘ Note

It is advised that the column of string field which is pre-populated should be removed from the import file. Else, the field value present in import file takes the precedence and overwrites the one specified in the rule.

As soon, the job is completed, you could see that the field on Manage Data page is automatically set to the value specified in initialize rule.

#### ⓘ Note

Only **Create** operation is allowed to call the initialize rules. Other data operation like **Insert/Update/Delete** does not trigger it.

Likewise, you can also assign the rules to composite child entities. Initialize Rules support composite child entities for both UI interaction and import.

## 10.6.2 Validate Rules

Validation rules ("validateRules") are triggered after a change to an object is submitted but before the change is saved (an "onValidate" event). You can use these rules to validate field values. For example, validation rules can be used to make sure that numbers cannot be entered in the field Employee Name. validateRules is used to set Error/Warning/Info message when rule condition is met. validateRule is triggered by CREATE/UPDATE/INSERT operation.

Once the validate rule is added to the object definition, you have to import an object with external name values which has an associated error defined in the validateRule linked to it.

Once Import job is completed, the error message specified in the rule gets logged in the import file.

Likewise, you can also assign the rules to composite child entities. Validate Rules support composite child entities for both UI interaction and import.

## 10.6.3 Save Rules

Save rules ("saveRules") are triggered when a user tries to save changes to an object (an "onSave" event). You can use these rules to check related field values for correctness. For example, a field could become required as a consequence of a save. saveRule can be triggered by CREATE/UPDATE/INSERT operation.

Once the save rule is added to the object definition, make some change on the object and save it. this change could be in form of CREATE/UPDATE/INSERT operation.

Once Import job is completed, you would see that the string field object on Manage Data page is automatically set to the value specified in the rule.

Likewise, you can also assign the rules to composite child entities. Save Rules support composite child entities for both UI interaction and import.

# 11 Integrating with Business Rules

## 11.1 Business Rules in MDF

You can assign business rules to MDF object definitions to add additional application logic.

Business rules are used to add application logic to determine the outcome of a change made to particular data in the system. A business rule scenario provides application-specific guidance about various objects, parameters, and actions you can use to configure the rule. To create rules for MDF objects, we recommend that you use the [Metadata Framework > Rules for MDF Based Objects](#) rule scenario to avoid potential errors.

### ⚠ Caution

Avoid using the *Basic* rule scenario, especially if you're doing a fresh implementation.

Follow the instruction in this section to create and apply business rules to MDF object definitions.

### Considerations for Workflow in Rules

- For **Basic Rule** scenarios, you can configure a Workflow. However, SAP recommends you to use **Rules for MDF Based Objects** scenario with **Workflow** purpose.
- With new rules scenario, you are only allowed to attach the workflow rules to Save.
- SAP recommends you to convert your workflow basic rules to the scenario-based rules and then attach as Save rule.

#### [Configuring a Business Rule for MDF \[page 154\]](#)

Use the MDF business rule scenario to create a business rule so that you can attach it to an MDF object definition.

#### [Assigning a Rule to an MDF Object \[page 158\]](#)

Assign an MDF-based rule to an object so that it can be triggered.

#### [Order of MDF-Based Rule Execution \[page 159\]](#)

Understand the order of rule execution during save in MDF objects to help you design and implement business rules in MDF.

### Related Information

[Implementing Business Rules in SAP SuccessFactors](#)

## 11.1.1 Configuring a Business Rule for MDF

Use the MDF business rule scenario to create a business rule so that you can attach it to an MDF object definition.

### Procedure

1. Go to [Admin Center](#) > [Configure Business Rules](#) , and choose + (Create New Rule).

You can also call up the create rule screen in [Configure Object Definitions](#) or [Extension Center](#) when you create or edit an object definition.

2. Expand [Metadata Framework](#) scenario and select the option [Rules for MDF Based Objects](#).

We don't recommend using the Basic scenario to create rules for MDF.

Attributes corresponding to the selected scenario appears on the UI.

3. In the [Rule Name](#) field, provide a name that can help you identify the rule.
4. In the [Rule ID](#) field, provide a unique Rule ID.

#### Note

The **Rule ID** field is automatically updated with the value you provided in **Rule Name** field.

5. In the [Start Date](#) field, specify the date from which you want the rule to be effective.
6. In the [Description](#) field, enter a description for your rule.
7. From the [Base Object](#) dropdown, select an object that you want to define a rule for.
8. From the [Purpose](#) dropdown, select the purpose of your rule.

When you create a rule scenario, you must state its purpose. Based on the selected purpose, the rule configuration displays certain field or attributes in the IF or ELSE statement. Each purpose can be attached to only specific events on the [Configure Object Definitions](#) page. These restrictions eliminate the possibility of incorrect configurations while defining or attaching a rule in a rule scenario. For more information about rule purposes, refer to the Related Information section.

9. Choose [Continue](#).


Details of the rule along with the added parameters are displayed on the UI.

The **Parameter** section displays auto-generated values based on the selection of Base Object and Purpose.

#### Note

Lock icon appears next to the parameters that are read-only.

If the selected Base Object is effective dated, then effective dated parameters are also displayed.

10. You can now define If-Then-Else configurations. For more information on how to configure rule conditions, please refer to the section [Implementing Business Rules in SAP SuccessFactors](#) > [IF, THEN, ELSE, and ELSE IF Statements](#) .

### Note

While defining business rules, you can set the value for the *External User Visibility* field for custom MDF objects. Currently, this is only applicable for Onboarding 2.0 use cases.

11. Save your changes.

## Results

Once you've configured the rule, it's available on the [Admin Center > Configure Business Rules](#) page.

## Next Steps

You can now assign the rule to an MDF object.

## Related Information

[Rule Parameters and Context \[page 157\]](#)

[Supported Purposes for MDF Rule Scenarios \[page 155\]](#)

### 11.1.1.1 Supported Purposes for MDF Rule Scenarios

While the MDF Rule Scenario allows you to configure rules with a purpose, not all purposes are supported for every event.

## Important Notes about Rule Usage

Take a look at the following guidelines before you configure a business rule for an MDF object:

- Do not use a save rule to **change** system fields such as `effectiveStartDate` and `effectiveEndDate`. Changing these fields using save rules can lead to data corruption and application errors. If you want to check the values of such fields, use a validate rule instead.
- If an `onChange` rule is assigned to a field that is either hidden or displayed due to the change of its conditional field value, then the associated `onChange` rules are triggered.
- If you use the Configurable UI tool to define the fields of an object, the *On Load* rules must also be attached to the configuration UI.

## Purposes of MDF Rule Scenario in Configure Business Rule

| Purpose    | Description  |
|------------|--|
| Initialize | To initialize fields with default values. Here you can set values for any field, including system and business fields. They're executed through the UI.  |
| Validate   | To validate field values and generate error or warning messages based on the values of one or more fields.   |
| Evaluate   | To assign values to fields based on other field values during the <b>Save</b> operation. For example, you can set allowed leaves based on selected location. However, you can't set values for system fields and business keys such as <i>External Code</i> or <i>Effective Start Date</i> . |
| Alert      | To generate MDF alerts. For more information, see <b>MDF Alerts</b> in the Related Information section.  |
| Workflow   | For handling workflows.  |
| On Load    | To perform operations or validations on page load.   |

## Supported Purposes on Object Level

Use this table to find out which rule purposes are supported in the events of an MDF object:

| Event      | Purpose    |          |          |       |          |             |
|------------|------------|----------|----------|-------|----------|-------------|
|            | Initialize | Validate | Evaluate | Alert | Workflow | Basic Rules |
| Initialize | Yes        | -        | -        | -     | -        | All         |
| Validate   | -          | Yes      | -        | -     | -        | All         |
| Save       | -          | Yes      | Yes      | -     | Yes      | All         |
| Post Save  | -          | Yes      | -        | Yes   | -        | All         |
| Delete     | -          | Yes      | -        | -     | Yes      | All         |

## Supported Purpose on Field Level

Use this table to find out which rule purposes are supported on the field level events of an MDF object:

| Event    | Purpose    |          |          |       |          |        |        |
|----------|------------|----------|----------|-------|----------|--------|--------|
|          | Initialize | Validate | Evaluate | Alert | Workflow | onLoad | Legacy |
| onChange | -          | Yes      | Yes      | -     | -        | -      | All    |

| Event  | Purpose           |
|--------|-------------------|
| onLoad | - - - - - Yes All |

## Related Information

[MDF Alerts \[page 194\]](#)

### 11.1.1.2 Rule Parameters and Context

Rules use parameters and contexts to help you define a rule and when you want the rule to be executed.

Rule parameters in MDF rule scenario:

- **Original Record** – If you want to compare changes made to the field values or if you want to identify the original field values in the system before executing a specific rule, you can use the Original Record rule parameter. Using the **Original Record** parameter, you can compare the current field value with the actual value of the field in the system before changes were made to it. This can only be used only an IF or Else-IF statements.

#### Note

During rule execution, the Original Record returns values for both the parent object and it's composite child.

- **Previous Effective Dated Record** – This parameter is applicable for objects that are effectively dated as Basic or Multiple Changes Per Day (MCPD). You can use this parameter to compare values with previous effective-dated records. For example, if you have three records in the system with effective dates January 1, 2015, July 15, 2015, and December 31, 2015. Now, if there was a change made to the record dated July 15, 2015, during the rule execution process, and you want to know the value of the record for January 1, 2015 (an earlier date), you can use the Previous Effective Dated Record. This can only be used only an IF or Else-IF statements.
- **Next Effective Dated Record** - This parameter is applicable for objects that are effectively dated as Basic or Multiple Changes Per Day (MCPD). You can use this parameter to compare values with previous effective-dated records. For example, if you have three records in the system with effective dates January 1, 2015, July 15, 2015, and December 31, 2015. Now, if there was a change made to the record dated July 15, 2015, during the rule execution process, and you want to know the value of the record for December 31, 2015 (a later date), you can use the Next Effective Dated Record. This can only be used only an IF or Else-IF statements.

## Rule Context

Rule contexts help you decide when you want a rule to be executed. For example, you can use rule contexts if you want to execute a rule using the user interface, but you want to exclude it during the import operation. This can be used only in an IF or Else-IF statements.



Currently, the following rule contexts are supported:

- **Data Operation.** Use this rule context if you would like to execute rules to perform an operation. For example, you can use this rule context to execute rules to create, insert, or make a correction for a specific record but would like to exclude the delete operation.
- **Enable on Import.** Use this rule context if you want a rule to be executed during import or not. In an IF or Else-IF statement, use **No** if you do not want to execute the rule during import and **Yes** if you want to execute the rule during import.  
By default, the option displayed is **Yes** if this rule context is not used.
- **Is DRTM Run.** Use this rule context when a rule must be executed during DRTM purge.

#### Note

Some rules don't get executed during the **Save** operation. For more information, see **Rule Integration with MDF Import** in the Related Information section.

## Workflow Context

Workflow contexts are applicable only with **Workflow** purpose. You must use this context parameter if you want to configure an object to trigger workflows based on rules. This can be used only in a THEN or ELSE statement and the **Workflow Configuration** value will decide the workflow that needs to be triggered.

## Related Information

[Rule Integration with MDF Import \[page 151\]](#)

## 11.1.2 Assigning a Rule to an MDF Object

Assign an MDF-based rule to an object so that it can be triggered.

### Context

You can check whether a rule is assigned to an object in [Configure Business Rules](#). To assign a rule to an MDF object, you can go to either [Configure Object Definitions](#) or [Extension Center](#) and modify the object definition. This topic provides instructions on how to do it in [Configure Object Definitions](#).

## Procedure

1. Go to [Admin Center](#) > [Configure Object Definitions](#).
2. From the *Search* dropdown, select the option *Object Definition*.
3. From the second dropdown, select the object definition for which you created the rule.
4. Click [Take Action](#) > [Make Correction](#).
5. In the *Rules* section, select the rule you created.

Based on the purpose you specified, the created rule appears under one of the rule type dropdowns.

6. Click [Save](#).

## Next Steps

Check that the rule is assigned to an Object in [Admin Center](#) > [Configure Business Rules](#) page. Search for your rule and check the **Assigned** column. If the check mark appears, your rule is assigned.

### 11.1.3 Order of MDF-Based Rule Execution

Understand the order of rule execution during save in MDF objects to help you design and implement business rules in MDF.

The triggering of business rules during save takes the following order:

1. Triggering of *Basic* rules assigned to objects as Validate Rules in order of assignment.
2. Triggering of *MDF Rule Scenario* rules assigned as Save Rules with workflow purpose in order of assignment.

#### → Remember

Currently, the evaluation and calculation of which workflows are triggered are determined during the **triggering of workflows** stage (stage 6 in the order). It minimizes the undesired behavior of workflow determination based on fields that are changed during Save rule execution.

#### ⚠ Caution

We don't recommend using multiple workflow rules or combining rules of Basic scenario and MDF Rule Scenario for workflows. These rules can run into issues.

3. Triggering of all *Basic* rules assigned as Save Rules in order of assignment.
4. Triggering of all *MDF Rule Scenario* rules assigned as Save Rules in order of assignment.
5. Triggering of all *MDF Rule Scenario* rules assigned as Validate Rules in order of assignment.
6. Triggering of workflows.
7. Triggering of all *Basic* rules assigned as Post Save Rules in order of assignment.
8. Triggering of all *MDF Rule Scenario* rules assigned as Post Save Rules including Alert Rules in order of assignment.

## 11.2 Editing the Rule Type Picklist

This section explains how to use MDF and the rules engine to add custom rules to your MDF instance.

### Context

The rules engine is a particular type of generic object that is created using MDF. You can use rules to add simple business logic to your instance. For MDF, rules are defined in the picklist *RuleType*. You can edit this picklist to add custom rules.

### Procedure

1. Go to [Picklist Center](#).
2. In the search field, type "rule".
3. In the search result, choose picklist [RuleType](#).
4. To add new key value pairs for rules for your instance, go to the active picklist and choose + ([Add New Value](#)).
5. On the [Add New Value](#) screen, enter the following information:

| Field                | Description   |
|----------------------|---|
| <b>External Code</b> | Unique external code for the rule.                      |
| <b>Label</b>         | Name by which you want to refer to this rule in the UI. |
| <b>Status</b>        | To activate the rule, set to active.                    |

6. Choose [Create](#). After you have saved this rule type, it will be available for you to configure on the [Configure Business Rules](#) page.

### Related Information

[Configuring a Business Rule for MDF \[page 154\]](#)

## 11.3 Rules for Composite Child Entities

This section explains rules for composite child entities.

You can run rules for composite child entities. In Employee Central, most of the objects related to employment and person are composite objects. The parent object has external code of type *User* or *Person*, with a one-to-many relationship with a composite object containing the details (child).

MDF allows users to assign rules on initialize, validate, save, and delete events. While all these events are triggered for the parent objects, they are omitted for the composite child entities at this point. This is because the four types of rules are executed in the corresponding rules processor methods, and processors are not executed for composite child entities.

### Note

To prevent incorrect assignment of rules, you must filter the available rules list on the UI by base object value to match the current object definition.

### Validate and Save Rules

Validate and Save rules are fully supported for composite child entities for both UI interaction and import.

### Delete Rules

Delete rules are supported for composite child entities if the composite child entity is deleted through the UI. If you delete the parent entity, the Delete rules are not executed. Note that the Delete rules are not executed during import.

The processors stack is implemented in such a way that there is no easy way to run processors for composite child entities. As a temporary fix for rules, you can initiate rules execution for the child entities from the context of the parent entity, that is, the rules processor methods executed for the parent. This means that for each composite child of the parent entity, you must do the following:

1. Execute the Validate rules assigned to the child entity after executing the parent's Validate rules in the rules processor.
2. Execute the Save rules assigned to the child entity after executing the parent's Save rules in the rules processor.
3. Execute the Delete rules assigned to the child entity after executing the parent's Save rules for entities that must be removed. You must do this before executing the parent's Delete rules in the rules processor.

## Initialize Rules

Initialize rules are supported for composite child entities for both UI interaction and import. The fields of composite child objects are initialized with default values.

You must execute the Initialize rules for composite child entities when they are initialized. This means that you must do the following:

1. Execute the Initialize rules for a fake composite child instance in the `CompositeAssocDefVo` and pass the default values to the `DataGridVo`.
2. Execute the Initialize rules for a fake composite child instance and assign default values to the `GENERIC_OBJECT_T_STAGING` table during import.

## 11.4 Assigning Rule Execution Criteria for an MDF Object

Assign rule execution criteria for a business rule applied to an MDF object.

### Prerequisites

- You have the [▶ Metadata Framework > Manage Data](#) permission enabled.
- You have the [▶ Metadata Framework > Access to non-secured objects](#) permission enabled.
- You must have an object created for the *Object Configuration* object in *Manage Data*.

### Context

You want to execute business rules for all records in the parent object and its composite child object, or you want to execute business rules only for records that were added or updated in the parent object or in the composite child object.

### Procedure

1. Go to [▶ Manage Data > Object Configuration](#).
2. Search for the object for which you like to assign the rule execution criteria.
3. In the *Rule Execution Criteria* field select a desired criteria.

You have the following criteria to choose from:

- *Execute rules for all records*: Choose this option to apply the rule for all records in the parent object and its composite child objects.

- *Execute rules for updated records*: Choose this option to apply the rule only for updated records in the object definition.

#### → Remember

We recommend that you choose the *Execute rules for updated records* option to execute business rules only for updated records in an object configuration. Choosing *Execute rules for all records* in this scenario would impact the performance during *Save*. However, there will be no impact when you create or insert new records. In such cases, rules will be always executed for all records in the object and its composite children.

4. Choose *Save*.

## 11.5 Importing, Exporting, and Transporting Rules

### 11.5.1 Exporting Rules

You can copy rules from one company instance to another using the MDF import and export framework. Example, copying rules from the test instance to production instance.

#### ⓘ Note

Before copying rules, please ensure that the Object Definitions, Language Package, and Picklists used in rules must be in sync. If the base object of the rule is an Employee Central object based on the succession data model, you must ensure to sync up the related data model before copying rules.

You can associate rules with MDF objects so that they are triggered at different object events such as on initialize, save, postsave, validate, or delete, or on individual field events such as on field change.

To export rules based on generic objects:

1. Go to the [Admin Center > Import and Export Data](#).
2. In the *Import and Export Data* page, from the *Select action to perform* dropdown list, choose *Export Data*.
3. Specify the value for the fields that appear on UI.

| Field                     | Description   |
|---------------------------|---|
| Select Generic Object     | Select <i>Rules</i> .   |
| Include dependencies      | Choose <i>Yes</i> .   |
| Include immutable IDs     | It is recommended to leave this field to its default value of <i>No</i> |
| Exclude reference objects | It is recommended to leave this field to its default value of <i>No</i> |

| Field                   | Description   |
|-------------------------|---|
| Select all data records | <p>Choose <i>No</i>.</p> <p>An additional <i>Select objects</i> field appears, where you can select the rules of your choice.</p> <p>Choose <i>Yes</i>.</p> <p>To export all the rules.</p> |

4. Click *Export*.

## Results

An export job is triggered and added to the Job Scheduler queue. Check the result in *Scheduled Job Manager*. Once the export job is completed, you can download the object definition package using the *Download Status* link in *View Details*. The object definition package is a zip file with an individual CSV file for each individual object definition entities structure.

## 11.5.2 Importing Rule-Related Generic Objects

Once you export both the rules and object definitions from the source instance, you must import them to the target instance.

To import rule-related generic objects:

1. Go to ► *Admin Center* ► *Import and Export Data* ►.
2. In the *Import and Export Data* page, from the *Select action to perform* dropdown list, choose *Import Data*.
3. In the form, choose the file type and the required generic object package.
4. Click *Import*.

## Results

An import job is triggered and added to the Job Scheduler queue. Check the result in *Scheduled Job Manager*. Once the import job is completed, you can verify whether the rule-related generic objects have been imported by going to the *Configure Object Definition* page.

## 11.5.3 Importing Rules

You can import rules that you have previously exported from a source instance.

### Procedure

1. Go to ► [Admin Center](#) ► [Import and Export Data](#) ►.
2. On the [Import and Export Data](#) page, from the *Select action to perform* dropdown list, choose [Import Data](#).
3. In the form, choose the tab for the file type you want to import.
4. If you import a CSV file, select [Rule](#) in the *Select Generic Object* field, and select the purge type.

#### Note

If you use an import file that was extracted from a source instance before the 2H 2022 upgrade, you must use the purge type [Full Purge](#). If you want to use the purge type [Incremental Load](#), make sure that the import file contains all fields of the technical rule definition existing in the current release. For example, in 2H 2022 the field [legacyPicklistOptionMode](#) was added to the technical rule definition.

5. Choose the file.
6. Choose [Import](#).

### Results

An import job is triggered and added to the Job Scheduler queue. Check the result in [Scheduled Job Manager](#). Once the import job is completed, you can verify whether the rule-related generic objects have been imported by going to the [Configure Object Definition](#) page. Once both the rules and object definition packages have been imported, the rules are migrated from the source instance to the target instance.

### Next Steps

If you've imported a file that was extracted from a source instance before the 2H 2022 upgrade, you have to follow these steps:

- If the rules use option IDs for legacy picklist values, manually adjust the option IDs on the [Configure Business Rules](#) page. Else the rules might not deliver the results you expect.
- Run the [RuleTransportStabilization](#) migration check in the check tool. This check finds the rules that don't fulfill the current technical requirements to be stable in transport and import. Use the [Quick Fix](#) to migrate the rules to the current technical requirements to be stable in transport and import.



## Note

Running the migration updates the *updated by (last modified by)* information with the user who executes the *Quick Fix*. The *last modified on* information is updated with the time stamp of when the migration was performed.

Rules that contain configuration errors can't be migrated automatically with the quick fix. For these rules, open the results of the check in the *Check Tool* and select the corresponding result. The *Configure Business Rules* page opens with the corresponding rule, and the system automatically checks the rule for any potential errors. Manually correct the errors and save the rule, which automatically saves your correction and also automatically adapts the rule to fulfill the technical requirements for transport and import.

Adopting the rule to be stable for transport and import doesn't change your configuration of the rule.

## 11.5.4 Constraints on Rules Imports

This section gives an overview on how to avoid a failing rule import.

Import can fail in the following scenarios:

### Effective Start Date

You already have a rule type whose effective start date is later than the effective start date of the rule type in the importing package, and the latter rule type doesn't have the value for the rule that is to be imported.

### Example

In the following figure, the effective start date of the rule type is *01/01/2000* and it has only the value *type1*.

The screenshot displays two panels. The 'History' panel on the left shows a record for 'January 1, 2000' with the following details: Added: type1 (type1), External Code: type1, Label: type1, Status: Active, Deleted: type2 (type2). The 'Picklist' panel on the right shows a form for 'RuleType' with fields for Name, Status (Active), and Effective Start Date (01/01/2000 01:01:00). Below the form is a table with the following data:

| External Code | Label | Status |
|---------------|-------|--------|
| type1         | type1 | Active |

In the following figure, the effective start date of the rule type in the importing package is *01/01/1900* and it has the value *type2*. This value is for the rule that is to be imported.

The rule cannot be imported because the system cannot find type2. This is because it is not currently effective. You need to go into the rule and configure the rule type with only one record with the effective start date *01/01/1900*.

## Legacy Platform Picklists

Previously, there was a constraint on importing rules that used legacy picklist values. If the base object of the rule was an Employee Central entity such as job information, and the rule used a legacy picklist field, the option ID of the picklist value was stored. As the option ID is different from instance to instance, you had to correct the corresponding rules in the target system to replace the source system picklist value option ID with the target system picklist value option ID.

This constraint is no longer valid. Instead of the option ID of the legacy picklist value, the external code of the picklist value is stored. As the external code is unique for all instances, no additional maintenance of the rules after importing in the target system is necessary. However, if you import old rules that still use option IDs for legacy picklist values, follow the steps mentioned in *Importing Rules*.

## 11.5.5 Adding a Rule to a Transport Bundle

Add a rule to a transport bundle so that you can transport it to a paired system using the Configuration Transport Center.

### Prerequisites

You've created a transport bundle for the corresponding MDF object in Configuration Transport Center.

### Context

You can use bundles to transport the configuration of a source system to a paired target system so that you don't need to manually configure it.

## Procedure

1. Go to ► [Admin Center](#) ► [Manage Data](#) ► and open a rule.
2. Select ► [Take Action](#) ► [Add to Transport Bundle](#) ► to add the selected rule to an existing bundle.  
A list of available transport bundles are displayed.
3. Select the bundle you want to add the configuration to and select [Save](#).  
Your configuration is successfully added to the transport bundle. A success message is displayed.  
If the rule has a Used Message Definition, then this is added to the transport bundle with the rule.
4. Select [Close](#).

## Results

The rule is added to the transport bundle.

## 11.6 MDF Rule Scenario Use Cases

Possible use cases for MDF rule scenarios.

### [Providing Default Values \[page 169\]](#)

You can define the default value for an object's instance, by choosing the *Initialize* purpose. This sets the initial value for an object instance.

### [Validating User Input Text or Values \[page 170\]](#)

You can validate field values based on the input value entered for one or more fields.

### [Setting Field Values Based on User Input \[page 172\]](#)

You can automatically assign values to fields by evaluating other field values entered by the user, and then, saving these field values.

### [Generating Alert Notifications \[page 173\]](#)

You can trigger alerts and notifications for any MDF object to proactively inform users about a business event and enable prompt actions.

### [Configuring Rule-Based Workflow \[page 175\]](#)

Workflows are approval processes initiated by any user to make changes to an employee's data or to an object. While you can configure a workflow using basic rules, we recommend that you use the Rules for MDF-Based Objects with Workflow as the purpose.

### [Populating Calculable Fields on the User Interface \[page 177\]](#)

Transient field values are not stored in the database because they are populated at runtime on the user interface (UI) based on rules.

### [Updating Field Values Based on Changes on Another Field \[page 179\]](#)

You can define rules to trigger automatic updates in one field when the value of an existing field is updated.

### [Preventing Deletion of Specific Records \[page 180\]](#)

You can set rules to prevent the deletion of specific records.

### [Changing Field Attributes on the User Interface \[page 181\]](#)

Use the Configurable UI to change field attributes on the user interface.

### [Excluding Rule Execution during Import \[page 183\]](#)

You can choose to exclude a rule from being executed during the import operation.

### [Validate Rules Based on Next, Previous, or Original Values \[page 184\]](#)

This use case is applicable for *Effective Dated Basic* objects and *Multiple Changes Per Day* objects to compare new values against old values. You can also compare current values with the previous or next effective dated record.

## 11.6.1 Providing Default Values

You can define the default value for an object's instance, by choosing the *Initialize* purpose. This sets the initial value for an object instance.

### Context

For example, consider an effective dated object **User Training Information**. Consider the *Language* field on the picklist for which you'd like to assign the default value as English. You can also use the Initialize purpose to set the default start date of any training to be the 1st of June 2019.

**Defaulting Rule (Defaulting\_Rule)**  
Scenario: Rules for MDF Based Objects

**Basic Information**

|             |            |
|-------------|------------|
| Start Date  | 01/01/1900 |
| Description |            |
| Base Object | cust_basic |
| Purpose     | Initialize |

**Parameters**

| Name       | Object         |
|------------|----------------|
| Context    | System Context |
| cust_basic | cust_basic     |

**Then**

- Set cust\_basic.lang to be equal to English (E1T\_English)
- Set cust\_basic.effectiveStartDate to be equal to Get First Day Of Month() Date: Today()

**Security**

- Secured: No
- Permission Category: No Selection
- RBP Subject User Field: ?
- CREATE Respects Target Criteria: No
- Base Date Field For Blocking: ?

**Rules**

- Initialize Rules: Defaulting Rule (Defaulting\_Rule)

**Validate Rules**

No data for Validate Rules available or you do not have permission to view this data.

**Save Rules**

No data for Save Rules available or you do not have permission to view this data.

**cust\_basic:**

- effectiveStartDate: 06/01/2019
- externalCode: Click or focus to edit
- externalName: Click or focus to edit
- lang: English (E1T\_English)

## Note

Initialize rules are only executed for new object instances. They will not be executed if you update the object's instance, insert a new record, or delete the object. Also, the rule will be executed before the UI loads completely.

## Procedure

1. Go to ► [Admin Center](#) ► [Configure Business Rules](#) ► page, and choose + to create new Rules UI.
2. Expand [Metadata Framework](#) scenario and select the option [Rules for MDF Based Objects](#).
3. Enter the rule details such as [Rule Name](#), [Rule ID](#), [Start Date](#), and [Description](#) in the respective fields.
4. From the [Base Object](#) dropdown, select an object that you want to define a rule for.
5. From the Purpose dropdown, select the purpose of your rule as [Initialise](#) and set your default value for fields of your object.
6. Add business logic and save the Rule.
7. Attach this rule scenario in the [Initialize Rules](#) section in the ► [Configure object definition](#) ► [Make Correction](#) ► field and [Save](#) the changes.

## 11.6.2 Validating User Input Text or Values

You can validate field values based on the input value entered for one or more fields.

### Context

Use the [Rules for MDF Based Objects](#) scenario to choose the [Validate](#) purpose and enter the required input value for your object. You must then, attach this rule scenario in the [Validate Rules](#) section in the Object Definition field.

For example, when you change your marital status to [Married](#), the application will prompt you to enter your spouse's name and your anniversary date. By choosing the field value as [Married](#), you are validating your choice by entering supporting input values of spouse name and anniversary date.

● **Benefits Form (Benefits\_Form1)**

Scenario: Rules for MDF Based Objects

Basic Information

Start Date: 01/01/1900

Description: Benefits Form

Base Object: Benefits Form

Purpose: Validate

| Name            | Object         |
|-----------------|----------------|
| Context         | System Context |
| Benefits Form   | Benefits Form  |
| Original Record | Benefits Form  |
| Rule Context    | Rules Context  |

Variables

If

- Benefits Form.Marital Status is equal to Married (M)
- and
  - or
    - Benefits Form.Spouse Name is equal to Null
    - Benefits Form.Anniversary Date is equal to Null

Then

Raise Message " MI " with Error severity  
Please fill in Spouse Name and Anniversary Date.

Rules

Initialize Rules: No data for Initialize Rules available or you do not have the necessary permission.

Validate Rules: Benefits Form (Benefits\_Form1)

Save Rules: No data for Save Rules available or you do not have the necessary permission.

Post Save Rules: No data for Post Save Rules available or you do not have the necessary permission.

Delete Rules: No data for Delete Rules available or you do not have the necessary permission.

Benefits Form:

externalCode: Narendra Lahoti

Marital Status: Married (M)

Spouse Name: Click on focus to edit

Anniversary Date: Error

Please fill in Spouse Name and Anniversary Date.

Cancel Save

OK

## Procedure

1. Go to ► [Admin Center](#) ► [Configure Business Rules](#) ► page, and choose + to create new Rules UI.
2. Expand [Metadata Framework](#) scenario and select the option [Rules for MDF Based Objects](#).
3. Enter the rule details such as [Rule Name](#), [Rule ID](#), [Start Date](#), and [Description](#) in the respective fields.
4. From the [Base Object](#) dropdown, select an object that you want to define a rule for.
5. From the Purpose dropdown, select the purpose of your rule as [Validate](#) and enter the required input value for your object.
6. Add business logic and save the Rule.
7. Attach this rule scenario in the [Validate Rules](#) section in the ► [Configure object definition](#) ► [Make Correction](#) ► field and [Save](#) the changes.

## 11.6.3 Setting Field Values Based on User Input

You can automatically assign values to fields by evaluating other field values entered by the user, and then, saving these field values.

### Context

Use the *Rules for MDF Based Objects* scenario to choose the *Evaluate* purpose and enter the required values to your object. You must then, attach this rules scenario in the *Save Rules* section in the Object Definition field.

For example, the *Benefits Form* has the following fields – *Marital Status*, *Spouse Name*, and *Anniversary Date*. You can populate the *Max Allowance* field based on the values entered for Date of Birth (DOB) and Marital Status.

The image shows a screenshot of the 'Calculate Allowance' rule configuration interface. The rule is named 'Calculate Allowance (Calculate\_Allowance)' and is configured for the 'Benefits Form' object. The rule logic is as follows:

- Variables:**
  - If:**
    - Benefits Form.Date Of Birth is not equal to Null
    - Benefits Form.Date Of Birth is on or before 01/01/1990
    - Benefits Form.Marital Status is equal to Married (M)
  - Then:** Set Benefits Form.Max Allowance to be equal to 5000
  - Else:** Set Benefits Form.Max Allowance to be equal to 1000

The 'Save Rules' section shows the rule is attached to the 'Benefits Form (Benefits\_Form1)' object. The 'Post Save Rules' section shows the rule is applied to the 'Benefits Form: Yuri Miles (Bill)' object.

The 'Benefits Form: Yuri Miles (Bill)' form shows the following values:

- externalCode: Yuri Miles
- Marital Status: Married (M)
- Spouse Name: Jennifer
- Anniversary Date: 06/05/2015
- Date Of Birth: 06/01/1989
- Max Allowance: 5000

### Procedure

1. Go to **Admin Center** > **Configure Business Rules** page, and choose **+** to create new Rules UI.
2. Expand *Metadata Framework* scenario and select the option *Rules for MDF Based Objects*.
3. Enter the rule details such as *Rule Name*, *Rule ID*, *Start Date*, and *Description* in the respective fields.
4. From the *Base Object* dropdown, select an object that you want to define a rule for.

5. From the Purpose dropdown, select the purpose of your rule as *Evaluate* and enter the required values to your object.
6. Add business logic and save the Rule.
7. Attach this rule scenario in the *Save Rules* section in the ► *Configure object definition* ► *Make Correction* ► field and *Save* the changes.

#### ⓘ Note

If the user leaves the Max Allowance field blank or even enters a value, the value will be changed to a value determined by the rule.

## 11.6.4 Generating Alert Notifications

You can trigger alerts and notifications for any MDF object to proactively inform users about a business event and enable prompt actions.

### Context

Use the *Rules for MDF Based Objects* scenario to choose the *Alert* purpose and enter the required values to your object. You must then, attach this rules scenario in the *Post Save Rules* section in the Object Definition field.

For example, let's assume that your driving license is due to expire in 30 days and is due to be renewed. Lets consider *Driving License* as a parameter has an *Expiry Date* field that needs to send out an alert when the value reaches 30 days or lesser.



## Procedure

1. Go to **Admin Center > Configure Business Rules** page, and choose **+** to create new Rules UI.
2. Expand **Metadata Framework** scenario and select the option **Rules for MDF Based Objects**.
3. Enter the rule details such as **Rule Name**, **Rule ID**, **Start Date**, and **Description** in the respective fields.
4. From the **Base Object** dropdown, select an object that you want to define a rule for.
5. From the **Purpose** dropdown, select the purpose of your rule as **Alert** and enter the required values to your object.
6. Add business logic and save the Rule.
7. Attach this rule scenario in the **Post Save Rules** section in the **Configure object definition > Make Correction** field and **Save** the changes.

## Related Information

[MDF Alerts \[page 194\]](#)

## 11.6.5 Configuring Rule-Based Workflow

Workflows are approval processes initiated by any user to make changes to an employee's data or to an object. While you can configure a workflow using basic rules, we recommend that you use the Rules for MDF-Based Objects with Workflow as the purpose.

### Context

#### ⚠ Caution

Do not create dependency on other rules in a workflow rule. Such dependencies will likely cause issues in workflow routing. For example, if a workflow rule is determined based on the "Department" field, but the value of the "Department" field is changed by another rule. This workflow rule then has dependency on the other rule.

Use the *Rules for MDF Based Objects* scenario to choose the *Workflow* purpose and trigger the next stage of the workflow process. You must then, attach this rules scenario in the *Save Rules* section in the Object Definition field. Based on the value of the *Max Allowance* field, any one of the workflow actions associated with the rule is triggered.

For example, the *Benefits Form* has the following fields – *Marital Status*, *Spouse Name*, and *Anniversary Date*. You can trigger a different workflow based on the value entered for the *Max Allowance* field.

## ✔ Benefits Workflow Rule (Benefits\_Workflow\_Rule)

Scenario: Rules for MDF Based Objects

### Basic Information



**Start Date** 01/01/1900

**Description**

**Base Object** Benefits Form

**Purpose** Workflow

### Parameters

| Name             | Object   |
|------------------|--|
| Context          | System Context  |
| Benefits Form    | Benefits Form   |
| Workflow Context | Workflow Context   |

Show More

### Variables

#### If

Benefits Form.Max Allowance >= 5000

Then

Set Workflow Context.Workflow Configuration to be equal to High Value Benefits Workflow(BEN1)

#### Else If

Benefits Form.Max Allowance >= 2500

Then

Set Workflow Context.Workflow Configuration to be equal to Normal Benefits Workflow(BEN2)

#### Else

Set Workflow Context.Workflow Configuration to be equal to Default Benefits Workflow(WF10)

## Procedure

1. Go to **Admin Center** > **Configure Business Rules** page, and choose **+** to create new Rules UI.
2. Expand **Metadata Framework** scenario and select the option **Rules for MDF Based Objects**.
3. Enter the rule details such as **Rule Name**, **Rule ID**, **Start Date**, and **Description** in the respective fields.
4. From the **Base Object** dropdown, select an object that you want to define a rule for.
5. From the **Purpose** dropdown, select the purpose of your rule as **Workflow** to trigger the next stage of the workflow process.

### Note

You can trigger a different workflow based on the value entered for the **Max Allowance** field.

6. Add business logic and save the Rule.
7. Attach this rule scenario in the *Save Rules* section in the **Configure object definition** **Make Correction** field and *Save* the changes.

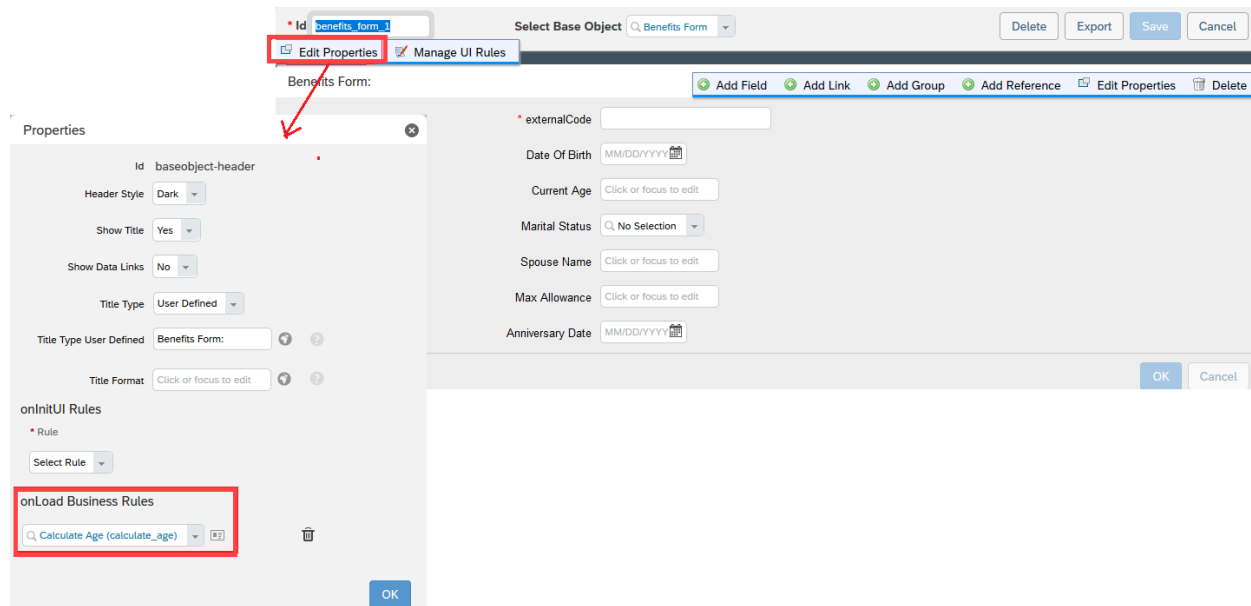
## 11.6.6 Populating Calculable Fields on the User Interface

Transient field values are not stored in the database because they are populated at runtime on the user interface (UI) based on rules.

### Context

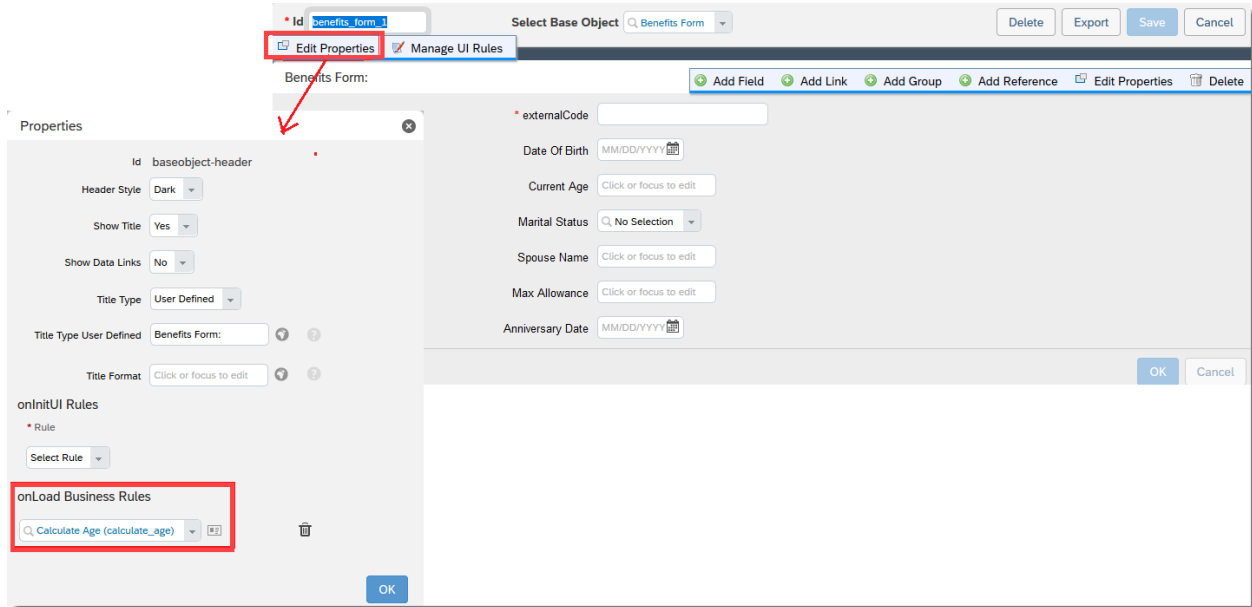
Use the *Rules for MDF Based Objects* scenario to choose the *On Load* purpose to calculate the field value. You must then, attach this rules scenario in the *On Load Rules* section in the Object Definition field.

For example, the *Benefits Form* has a *Date of Birth* field, which is a transient field that can help you calculate the age of the individual as of "today". By setting *Yes* to make the field transient, you are not storing the data, but displaying the value using simple calculations at runtime. Once the rules are attached and you load an instance with the *Date of Birth* field, the age of the individual is calculated automatically.



### Note

This use case is only applicable to the Manage Data page. If you are using the Configurable UI to define your fields, you must attach the same rule there as well.



## Procedure

1. Go to ► [Admin Center](#) ► [Configure Business Rules](#) ► page, and choose + to create new Rules UI.
2. Expand [Metadata Framework](#) scenario and select the option [Rules for MDF Based Objects](#).
3. Enter the rule details such as [Rule Name](#), [Rule ID](#), [Start Date](#), and [Description](#) in the respective fields.
4. From the [Base Object](#) dropdown, select an object that you want to define a rule for.
5. From the Purpose dropdown, select the purpose of your rule as [On Load](#) to calculate the field value.
6. Add business logic and save the Rule.
7. Attach this rule scenario in the [On Load Rules](#) section in the ► [Configure object definition](#) ► [Make Correction](#) ► field and [Save](#) the changes.

### Note

The rules are applicable to the UI only and must be attached on the [Manage Data](#) page when an object is configured.

# 11.6.7 Updating Field Values Based on Changes on Another Field

You can define rules to trigger automatic updates in one field when the value of an existing field is updated.

## Context

For example, the *Position* field is dependent on two other fields - *Job Title* and *Job Code*. Job Code is also a generic object that has *Job Title* as the field. Set the *Position* field as *Default*. So, when you enter a value for the Job Code, the Job Title field is automatically updated based on the Job Code's value. You can also set a condition wherein, if there is an existing Job Title, you can retain its value.

**Note**  
The rules must be added at the field level in this scenario.

**Job Code Change Rule (Job\_Code\_Change\_Rule)**

Scenario: Rules for MDF Based Objects

**Basic Information**  
Start Date: 01/01/1900  
Description: Position  
Base Object: Position  
Purpose: Evaluate

**Parameters**

| Name             | Object   |
|------------------|----------|
| incumbent        | Position |
| changeReason     | Position |
| description      | Co       |
| jobTitle         |          |
| jobCode          |          |
| jobLevel         |          |
| employeeClass    |          |
| regularTemporary |          |
| payGrade         |          |

**Variables**

**If**

Position.Job Title is not equal to Null

**Then**

Set Position.Job Title to be equal to Position.Job Code.Job Title.Default Value

**Details**

- Name: jobCode
- Database Field Name: sfFields.sfFieldB
- Maximum Length: 38
- Data Type: Generic Object
- Valid Values Source: JobClassification
- Hide Old Value: No
- Decimal Precision: Click or focus to edit
- Include Inactive Users: No
- UI Field Renderer: Click or focus to edit
- Transient: No
- Help Text: Click or focus to edit
- Private or Sensitive Information: No
- Show Trailing Zeros: No
- Visibility: Editable
- Status: Active
- Label: Click or focus to edit
- Cascade: None
- Inactivated By: No Selection

**Rules**

- External Code: Job Code Change Rule (Job\_Code...)

**Field Criteria**

| Source Field Name  | Destination Field Name | Default Destination Value | Status |
|--------------------|------------------------|---------------------------|--------|
| effectiveStartDate | effectiveStartDate     | 1900-01-01                | Active |

## Procedure

1. Go to [Admin Center](#) > [Configure Business Rules](#) page, and choose **+** to create new Rules UI.
2. Expand [Metadata Framework](#) scenario and select the option [Rules for MDF Based Objects](#).
3. Enter the rule details such as *Rule Name*, *Rule ID*, *Start Date*, and *Description* in the respective fields.
4. From the *Base Object* dropdown, select an object that you want to define a rule for.
5. From the Purpose dropdown, select the purpose of your rule as *Evaluate*.
6. Add business logic and save the Rule.
7. Attach this rule scenario in the *On Change* section in the [Configure object definition](#) > *Job Code* field and [Save](#) the changes.

## 11.6.8 Preventing Deletion of Specific Records

You can set rules to prevent the deletion of specific records.

### Context

Use the [Rules for MDF Based Objects](#) scenario to choose the *Validate* purpose and attach the rule scenario to the [Delete Rules](#) in the Object Definition field.

For example, in a Department, you can make sure that only the Head of Department can delete their department.

**Department Delete Validation (Department\_Delete\_Validation)** Insert New Record

Scenario: Rules for MDF Based Objects

| Basic Information |            | Parameters                |                |
|-------------------|------------|---------------------------|----------------|
| Start Date        | 01/01/1900 | Name                      | Object         |
| Description       |            | Context                   | System Context |
| Base Object       | Department | Department                | Department     |
| Purpose           | Validate   | Original Record           | Department     |
|                   |            | <a href="#">Show More</a> |                |

**Rules**

- Initialize Rules: No data for Initialize Rules available or you do not have the necessary permission.
- Validate Rules: No data for Validate Rules available or you do not have the necessary permission.
- Save Rules: No data for Save Rules available or you do not have the necessary permission.
- Post Save Rules: No data for Post Save Rules available or you do not have the necessary permission.
- Delete Rules: Department Delete Validation (Department\_Delete\_Validation)**

**Logic:**

**If**

- and
  - Department.Head of Department is not equal to Login User()
  - Department.Head of Department is not equal to Null
  - Rule.Context.Data operation is equal to Delete

**Then**

Raise Message " Dept\_del " with Error severity  
You can not delete Department if you are not head of the head of the department.

## → Remember

Running a Delete Rules validation will have an impact when you do a Full Purge import. Upon full purge import, all the existing records are deleted and re-created. So if you do a full purge import, you will not be able to delete any record unless the logged-in user is Head of Department.

## Procedure

1. Go to ► [Admin Center](#) ► [Configure Business Rules](#) ► page, and choose + to create new Rules UI.
2. Expand [Metadata Framework](#) scenario and select the option [Rules for MDF Based Objects](#).
3. Enter the rule details such as [Rule Name](#), [Rule ID](#), [Start Date](#), and [Description](#) in the respective fields.
4. From the [Base Object](#) dropdown, select an object that you want to define a rule for.
5. From the Purpose dropdown, select the purpose of your rule as [Validate](#).
6. Add business logic and save the Rule.
7. Attach this rule scenario in the [Delete Rules](#) section in the ► [Configure object definition](#) ► [Make Correction](#) ► field and [Save](#) the changes.

## 11.6.9 Changing Field Attributes on the User Interface

Use the Configurable UI to change field attributes on the user interface.

## Context

Only use UI rules in the following scenarios:

- Hide or show a field dynamically
- Mark a field as read-only, editable, mandatory, or optional dynamically

### ⚠ Caution

We recommend that you use business rules instead of UI rules to handle more complex data changes, and use RBP field override to control field access if you want to restrict the access based on roles. Rule trace and instance sync are not supported in UI rules.

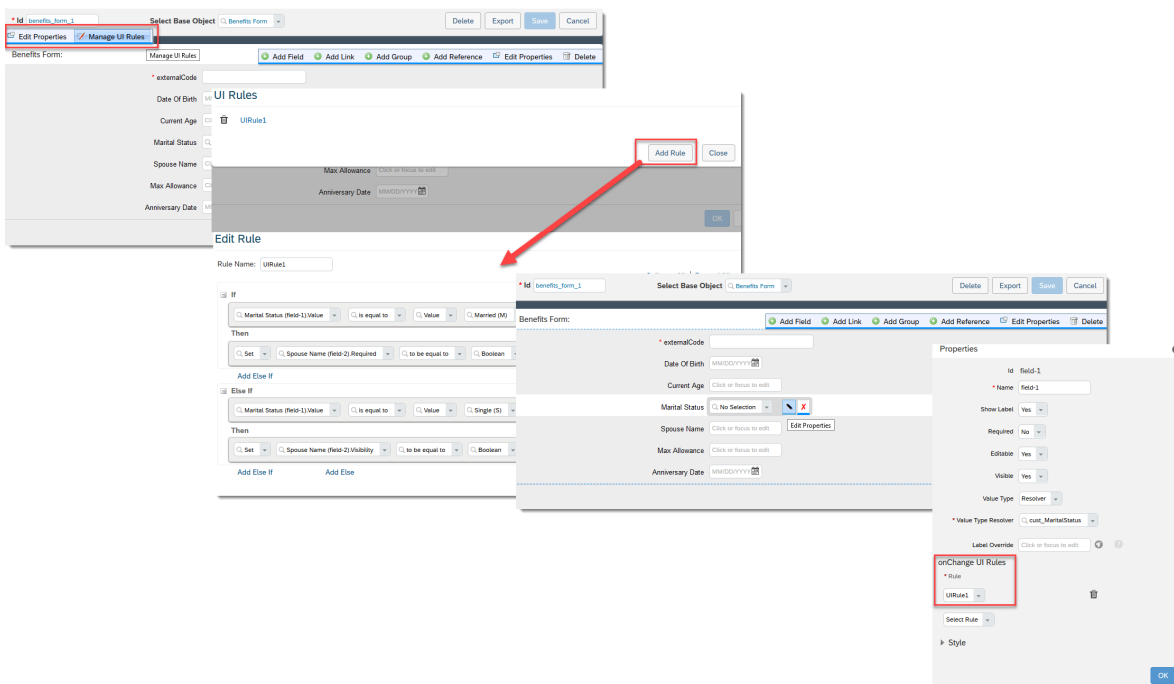
In the following example, the Benefits Form has the following fields – Marital Status, Spouse Name, and Anniversary Date. You can make the Spouse Name as mandatory if the Marital Status is set as Married. If the Marital Status is set to Single, the Spouse Name must be hidden. This can be done using Configurable UI and UI Rules.

To make sure that the Spouse field is set as Required or Hidden when the user enters a value for the Marital Status field, follow the steps below.



## Procedure

1. Create a Configurable UI.
2. You can also go to [Manage Configuration UI](#) and search for your configured UI.
3. Hover over the form and choose the option [Manage UI Rules](#).
4. On the UI Rules page, click [Add Rule](#). Keeping our example in mind, enter your rule criteria to make your [Spouse](#) field is required or hidden based on the value chosen at the [Marital Status](#) field. Click [Save](#) to save the rule.
5. Choose the field you want to attach the UI rules to and click the [Edit Properties](#) icon.
6. On the Properties page, go to the [onChange UI Rules](#) section and attach the rule. Click [OK](#).
7. Click [Save](#) to save the configurable UI.



## Related Information

[Creating a Configurable UI \[page 83\]](#)

## 11.6.10 Excluding Rule Execution during Import

You can choose to exclude a rule from being executed during the import operation.

### Context

Consider a scenario where a Cost Center that has a field Parent Cost Center. While creating a cost centre you have no information about the parent cost centre since it was created by another person. But you can set a default value (only on the UI) for the parent cost centre that can be updated later. While importing the cost centre, you do not want to apply this logic since the administrator is aware of the value to attach to each field and the import operation makes bulk updates for multiple records.

To do the above, you must set the rule context to *Enable on Import*. Use this rule context if you want a rule to be executed during import or not. In an IF or Else-IF statement, use *No* if you do not want to execute the rule during import.

The screenshot displays the configuration for a rule named "Parent Cost Center UI (Parent\_Cost\_Center\_UI)".

- Scenario:** Rules for MDF Based Objects
- Basic Information:**
  - Start Date: 01/01/1900
  - Description: Cost Center
  - Base Object: Cost Center
  - Purpose: Evaluate
- Parameters:**

| Name            | Object         |
|-----------------|----------------|
| Context         | System Context |
| Cost Center     | Cost Center    |
| Original Record | Cost Center    |
- Variables:**
  - and**
    - Cost Center.Parent Cost Center is equal to Null
    - Rule Context.Enable on import is equal to No
  - Then**
    - Set Cost Center.Parent Cost Center to be equal to To Be Decided (TBD)
- Rules:**
  - Initialize Rules: No data for Initialize Rules available or you do not have the necessary permission.
  - Validate Rules: No data for Validate Rules available or you do not have the necessary permission.
  - Save Rules: Parent Cost Center UI (Parent\_Cost\_Center\_UI)
  - Post Save Rules: No data for Post Save Rules available or you do not have the necessary permission.
  - Delete Rules: No data for Delete Rules available or you do not have the necessary permission.

## Procedure

1. Go to ► *Admin Center* ► *Configure Business Rules* ► page, and choose + to create new Rules UI.
2. Expand *Metadata Framework* scenario and select the option *Rules for MDF Based Objects*.
3. Enter the rule details such as *Rule Name*, *Rule ID*, *Start Date*, and *Description* in the respective fields.
4. From the *Base Object* dropdown, select an object that you want to define a rule for.
5. From the Purpose dropdown, select the purpose of your rule as *Evaluate*.
6. Add business logic and save the Rule.
7. Attach this rule scenario in the *On Change* section in the ► *Configure object definition* ► *Job Code* ► field and *Save* the changes.

### 11.6.11 Validate Rules Based on Next, Previous, or Original Values

This use case is applicable for *Effective Dated Basic* objects and *Multiple Changes Per Day* objects to compare new values against old values. You can also compare current values with the previous or next effective dated record.

Consider a scenario where a *Legal Entity* that has a field *Standard Weekly Hours*. Let's assume you want to set the following conditions:

- No warning message alert when a new Legal Entity is created.
- A warning message when Standard Weekly Hours have increased and there is no future effective dated record.
- A warning message when there is a change to Standard Weekly Hours and a record is available after that.

Using the *Rules for MDF Based Objects* scenario, choose the *Validate* purpose and attach the rule scenario to the *Validate Rules* in the Object Definition field. This rule will help you track changes made to the values for effective dated objects.

The screenshot displays the configuration for a rule named "Legal Entity Weekly Hrs Validation (Legal\_Entity\_Weekly\_hrs\_Validation)".

**Basic Information:**

- Start Date: 01/01/1900
- Description: Legal Entity
- Base Object: Legal Entity
- Purpose: Validate

**Parameters:**

| Name                            | Object         |
|---------------------------------|----------------|
| Context                         | System Context |
| Legal Entity                    | Legal Entity   |
| Original Record                 | Legal Entity   |
| Previous Effective Dated Rec... | Legal Entity   |
| Next Effective Dated Record     | Legal Entity   |

**Variables:**

- Legal Entity.Standard Weekly Hours > Previous Effective Dated Record.Standard Weekly Hours
- Previous Effective Dated Record is not equal to Null
- Next Effective Dated Record is equal to Null
- Next Effective Dated Record is not equal to Null
- Legal Entity.Standard Weekly Hours not equal to Original Record Standard Weekly Hours

**Then:**

- Raise Message "LE\_WH" with Warning severity: New weekly hours is more than earlier value
- Raise Message "LE\_WH\_NEXT\_EXISTS" with Warning severity: There is already a next record with value for Weekly Hours, it is not recommended to change the value

**History:**

- 01/01/2019: Standard Weekly Hours: 50 (Next Effective Dated Record)
- 01/01/2000: Legal Entity: Record created (Previous Effective Dated Record)

**Legal Entity: Abc Germany (ABCDEU) Details:**

- Effective as of: 01/01/2005 (Original Record Before any changes)
- Name: Abc Germany
- Status: Active
- Standard Weekly Hours: 40
- Country/Registration: Germany (DEU)

**Warning:** When a user tries to insert a new record between these two existing records.

### ⓘ Note

During rule execution, the Original Record returns values for both the parent object and it's composite child.

## 12 Workflows in MDF

Workflows are approval processes. It allows you to set up an approval process for changes that any user makes to an employee's data or to any other object. The system does not process the change until the approver approves the request.

As a system administrator, you can define which changes to an employee's data trigger a workflow, and who should approve the change. For example, you have to set up a workflow for changes to the national ID of an employee. If the user changes the national ID, a message appears indicating who has to approve this change. If the user confirms the change, the approver gets an approval request in the To-Do List and in the Pending request page. Once the workflow is approved, it is removed from the user's to do list. Typical usecases of workflows include:

- Leave approvals
- Candidate hiring processes
- Salary hikes

For MDF objects, if **Pending Data** is set to **Yes**, the data changes take effect only if approved. If the changes are not approved, the system rejects the changes. If **Pending Data** is set to **No**, it means that the data updates can be made without any approval in the workflow.

You can bulk remove all non-actionable records (to-be-approved records) if MDF records, with *Pending Data* set to *Yes*, has a pending workflow approval after a data change. Ensure that you have access to the check tool and have the required permissions to run checks from [Metadata Framework \(MDF\)](#) > [MDF Records with Workflow Issues](#). You can use the quick fix feature of the check tool to:

- Remove all non-actionable MDF records from *Manage Data* if there are no workflow requests associated with them. (*AssociatedWorkflowForNonActionableRecord*)
- Remove all non-actionable MDF records from *Manage Data* if the status of the associated workflow requests is not yet approved or rejected, and set their workflow status to cancelled. (*ValidWorkflowStatusForNonActionableRecord*)
- Regenerate to-do tasks for approvers on the home page for non-actionable records with a valid workflow. (*ToDoTaskForNonActionableRecord*)
- Remove MDF records with in-progress workflow status as these records can be edited from the *Manage Data* page, making them actionable. (*InProgressWorkflowWithApprovedRecord*)

### [Creating a New Workflow \[page 187\]](#)

You can define workflows to set up approval processes for changes made by a manager or employee to employee data. After you have created a workflow, you can associate it with an object.

### [Need to Know About Workflows in MDF \[page 188\]](#)

### [Supported Workflow Actions \[page 189\]](#)

Once the approver receives the workflow approval request, an action needs to be performed. For example, if the employee creates a leave request, the manager has can either approve it or decline it.

### [MDF Workflows on the Home Page \[page 190\]](#)

Metadata Framework Workflows include the following features on the home page.

## 12.1 Creating a New Workflow

You can define workflows to set up approval processes for changes made by a manager or employee to employee data. After you have created a workflow, you can associate it with an object.

### Context

As an Admin, you can create workflows using **Manage Organization, Pay and Job Structures** page.

#### Note

MDF workflows can be triggered for entities shown in *Manage Data* page but not from the *Configure Object Definition* page, for example, a workflow cannot be triggered while adding and deleting picklist values for an existing MDF picklist. MDF workflows do not support non-MDF picklists or picklists created on the *Configure Object Definition* page. An MDF picklist or MDF object definition creation is a super admin feature and MDF workflow doesn't apply to these two.

### Procedure

1. Go to ► [Admin Center](#) ► [Manage Organization, Pay and Job Structures](#) ► page.
2. From *Create New* dropdown, select option **Workflow**.

Workflow UI appears.

3. In *Workflow ID* field, provide a unique ID for your workflow.
4. In *Name* field, provide a name for your workflow.
5. In *Description* field, provide a description for your workflow.
6. In *Remind In Days* field, specify the number of days to set up reminders for your workflow objects.
7. From *Is Delegate Supported* dropdown, field to select Yes or No.

#### Note

**No** is the default selection.

8. If you want to trigger an alternate workflow, select the workflow from the *Alternate Workflow* dropdown.
9. From the *Redirect CC Users to Workflow Approval Page* dropdown, specify either **Yes** or **No**.

#### Note

Select **Yes**, when you want the individuals selected as CC Role redirected to the workflow approval page, so they can access workflow approval content. Select **No**, to allow individuals selected as CC Role can access the workflow object itself.

10. In the *Step* section, define any number of steps required in the approval process.

### Note

You can also define a workflow with no approval step.

11. In the *Contributors* section, add employees who do not act as approvers. Typically, contributors are notified about changes, and can provide comments.
12. In the *Cc Role* section, add employees who should be notified whenever a workflow is completed.
13. Click *Save*.

## Next Steps

Once the workflow is created, next step is to assign the workflow to an object. To know more on how to assign workflows, refer to section **Triggering Workflows** in [SAP SuccessFactors Employee Central > Implementing and Configuring Workflows in Employee Central](#) guide.

**Task overview:** [Workflows in MDF \[page 186\]](#)

## Related Information

[Need to Know About Workflows in MDF \[page 188\]](#)

[Supported Workflow Actions \[page 189\]](#)

[MDF Workflows on the Home Page \[page 190\]](#)

## 12.2 Need to Know About Workflows in MDF

- When you try to create, edit, or delete a record on a custom MDF portlet with a workflow attached to it, a link is displayed on portlet with the pending record and the effective date.

### Note

The link on the portlet allows you to view the records. You can't create or edit any records until the request is approved.

- When you delete an MDF record that has an associated workflow, an approval based on the workflow is required. A message on UI stating **This record has been marked for deletion. Deletion is pending approval** is displayed corresponding to the deleted object in **Manage Data** and **Manage Configuration UI** page.
- You can add multiple legacy rules to an MDF object that can trigger a workflow but only one of them is considered. No order is maintained for these attached rules, thus, it isn't known as to which rule takes precedence. SAP recommends using **Rules for MDF Based Objects** for workflow configuration for all MDF objects.
- MDF Rules Scenario allows you to add only one 'Workflow' rule to an MDF object. If you try to attach multiple workflow rules, an error appears.

### Note

MDF Rules Scenarios don't validate the attached legacy or basic rules.

- SAP recommends the use of MDF Rules Scenario for all rules related to MDF Objects including workflow.
- When Employee Central isn't enabled, you can approve MDF workflows but you can't access the [Workflow Details](#) page as a proxy user.

Parent topic: [Workflows in MDF \[page 186\]](#)

## Related Information

[Creating a New Workflow \[page 187\]](#)

[Supported Workflow Actions \[page 189\]](#)

[MDF Workflows on the Home Page \[page 190\]](#)

## 12.3 Supported Workflow Actions

Once the approver receives the workflow approval request, an action needs to be performed. For example, if the employee creates a leave request, the manager has can either approve it or decline it.

Supported Actions

| Workflow Action  | Description  |
|------------------|--|
| Approve          | The change is approved. The record is processed and the change appears in the system.  |
| Decline          | The change is declined. The request is not processed and no change appears in the system.  |
| Withdraw         | A request is withdrawn. No change appears in the system.   |
| Request          | A request is submitted and sent for approval.  |
| Comment          | Workflow participants add a comment to the request, which can be read by other participants.   |
| In-flight change | Approver makes minor changes to a request and then approves it. For example, an employee claims an insurance policy premium as part of IT Declarations and includes the late payment fee, which is not eligible. The administrator can approve only the basic insurance policy premium and edit the approved amount accordingly. |



| Workflow Action          | Description  |
|--------------------------|--|
| Send back (or push back) | Approver sends back a request so that the requestor can edit and re-submit the same request. This enables multiple use cases for IT Declarations, Reimbursements and Advances functionality. |

Parent topic: [Workflows in MDF \[page 186\]](#)

## Related Information

[Creating a New Workflow \[page 187\]](#)

[Need to Know About Workflows in MDF \[page 188\]](#)

[MDF Workflows on the Home Page \[page 190\]](#)

## 12.4 MDF Workflows on the Home Page

Metadata Framework Workflows include the following features on the home page.

### Note

- This home page content only appears if you've configured MDF objects with approval workflows.

| Name   | Description   | Where Shown   | When Shown  | Prerequisites | On Mobile App? |
|--|---|---|---|---------------|----------------|
| <ul style="list-style-type: none"> <li>MDF Workflow Requests</li> <li>MDF Workflow Requests - Sent Back</li> </ul> | <p>Enables you to approve pending MDF workflow requests, for changes to an MDF object that isn't employee-related. Links you to details.</p> <p>Initially, requests appear in the <a href="#">Approvals</a> section, where you can approve or decline. If a task is sent back by a subsequent approver in the workflow, it appears in the <a href="#">For You Today</a> section, so you can review and update it as necessary.</p> <p>(To-Do Category 17)</p> | <p><a href="#">Approvals</a> (for initial approval)</p> <p><a href="#">For You Today</a> (if sent back)</p> | <p>Appears initially when you have an MDF workflow request to approve. Appears again if the request is sent back by a subsequent approver in the workflow.</p> <p>Disappears when you approve or decline the request. Or, if no action is taken, it disappears 180 days after the due date (if set) or the Last Modified date (if there's no due date).</p> |               | No             |

**Note**

You can only approve the workflows on the home page. If you want to delegate, send back, or decline the workflows, you need to choose [View Details](#) to open the workflow details page.

| Name  | Description  | Where Shown  | When Shown  | Prerequisites | On Mobile App? |
|---|--|--|---|---------------|----------------|
| <ul style="list-style-type: none"> <li>MDF Workflow Requests - Employee-Related</li> <li>MDF Workflow Requests - Employee-Related, Sent Back</li> </ul> | <p>Enables you to approve pending MDF workflow requests, for changes to an MDF object that's employee-related. Links you to details. Initially, requests appear in the <a href="#">Approvals</a> section, where you can approve or decline. If a task is sent back by a subsequent approver in the workflow, it appears in the <a href="#">For You Today</a> section, so you can review and update it as necessary.</p> <p>(To-Do Category 17)</p> | <p><a href="#">Approvals</a> (for initial approval)<br/><a href="#">For You Today</a> (if sent back)</p> | <p>Appears when you have an MDF workflow request to approve. Appears again if the request is sent back by a subsequent approver in the workflow.</p> <p>Disappears when you approve or decline the request. Or, if no action is taken, it disappears 180 days after the due date (if set) or the Last Modified date (if there's no due date).</p> |               | No             |

### Note

You can only approve the workflows on the home page. If you want to delegate, send back, or decline the workflows, you need to choose [View Details](#) to open the workflow details page.

Parent topic: [Workflows in MDF \[page 186\]](#)

## Related Information

[Creating a New Workflow \[page 187\]](#)

[Need to Know About Workflows in MDF \[page 188\]](#)

[Supported Workflow Actions \[page 189\]](#)

# 13 MDF Alerts

You can define and trigger alerts and e-mail notifications for all the MDF objects including the custom objects. This helps you to proactively inform the employees, managers, or HR Admins upon a business event so that they can initiate the necessary activities at the right time.

Following is the implementation sequence to configure MDF Alerts:

1. Create an object in the [Configure Object Definitions](#) page.
2. Create records for the Object in [Manage Data](#) page.
3. Create an alert message in the [Manage Data](#) page so that the user gets a notification when the alert is triggered.
4. Create an instance for multiple alert type.
5. Create a basic workflow.
6. Configure the business rules to include the workflow.
7. Attach the rules for the object that you created in the [Configure Object Definition](#) page.

## 13.1 Configure MDF Alerts

Metadata Framework supports single alert and multiple alerts. Before you configure alerts, you need to ensure that you have defined a proper workflow and alert messages in the Rule. After you configure the rule, save the rule and attach this rule to the Object Definition.

Here is the detailed procedure on how to configure MDF Alerts:

### [Creating an Object Definition \[page 195\]](#)

By creating an object definition, it is possible for you to add information about fields, associations, workflow rules, and security for an object

### [Creating an Alert Message \[page 195\]](#)

In the [Manage Data](#) page, create an instance for the object (for example, `AlertMessage`) so that the user gets notified with a message when the alert is triggered.

### [Creating an Instance for Multiple Alerts \[page 196\]](#)

If you are configuring multiple alerts, you should create an instance for the object `MDFAlertType`. Otherwise, you can ignore this step.

### [Creating a Basic Workflow \[page 197\]](#)

Create a basic workflow to define the Approver and Cc role.

### [Configuring a Business Rule \[page 198\]](#)

The business rule basically consists of information that triggers the alert event.

### [Configuring Post Save Rules for an Object \[page 199\]](#)

To trigger the alert event so that the user gets the notification about any business event, it's required to post save the rule in the [Configure Object Definitions](#) page.

### [Creating Records for the Object \[page 200\]](#)

After you create and save the object, you should create records for the object that is already created in the [Configure Object Definition](#) page.

## 13.1.1 Creating an Object Definition

By creating an object definition, it is possible for you to add information about fields, associations, workflow rules, and security for an object

### Procedure

1. Go to ► [Admin Center](#) ► [Configure Object Definitions](#) ►.
2. Create a *Basic* dated object with externalCode data type as *User*. For example, cust\_Alert.  
For more information about creating a new object definition, see [Creating a New Object Definition \[page 19\]](#).

**Task overview:** [Configure MDF Alerts \[page 194\]](#)

### Related Information

[Creating an Alert Message \[page 195\]](#)  
[Creating an Instance for Multiple Alerts \[page 196\]](#)  
[Creating a Basic Workflow \[page 197\]](#)  
[Configuring a Business Rule \[page 198\]](#)  
[Configuring Post Save Rules for an Object \[page 199\]](#)  
[Creating Records for the Object \[page 200\]](#)

## 13.1.2 Creating an Alert Message

In the [Manage Data](#) page, create an instance for the object (for example, AlertMessage) so that the user gets notified with a message when the alert is triggered.

### Procedure

1. Go to ► [Admin Center](#) ► [Manage Data](#) ►.
2. Select [Alert Message](#) from the [Create New](#) dropdown.

3. Enter the values for each of the fields and save the object.

Following is the sample Alert Message configuration:

| Field Name            | Sample Configuration  |
|-----------------------|---|
| Name                  | AlertMessage1   |
| External Code         | AlertMessage1   |
| Effective Status      | Active  |
| Alert Header          | Rule1 Alert for [[SUBJECT_USER]]  |
| Alert Description     | Alert for [[SUBJECT_USER]], on [[EFFECTIVE_DATE]]                         |
| Localized Header      | Rule1 Alert for [[SUBJECT_USER]], [[OBJECT_NAME]],<br>[[OBJECT_TYPE]]     |
| Localized Description | Alert for Rule1 for [[SUBJECT_USER]], [[OBJECT_NAME]],<br>[[OBJECT_TYPE]] |

**Task overview:** [Configure MDF Alerts \[page 194\]](#)

## Related Information

- [Creating an Object Definition \[page 195\]](#)
- [Creating an Instance for Multiple Alerts \[page 196\]](#)
- [Creating a Basic Workflow \[page 197\]](#)
- [Configuring a Business Rule \[page 198\]](#)
- [Configuring Post Save Rules for an Object \[page 199\]](#)
- [Creating Records for the Object \[page 200\]](#)

### 13.1.3 Creating an Instance for Multiple Alerts

If you are configuring multiple alerts, you should create an instance for the object *MDFAlertType*. Otherwise, you can ignore this step.

## Procedure

1. Go to [Admin Center](#) > [Manage Data](#) >

2. Select *MDFAlertType* from the *Create New* dropdown.
3. Enter the *externalCode* and *externalName* and click *Save*.

**Task overview:** [Configure MDF Alerts \[page 194\]](#)

## Related Information

[Creating an Object Definition \[page 195\]](#)

[Creating an Alert Message \[page 195\]](#)

[Creating a Basic Workflow \[page 197\]](#)

[Configuring a Business Rule \[page 198\]](#)

[Configuring Post Save Rules for an Object \[page 199\]](#)

[Creating Records for the Object \[page 200\]](#)

## 13.1.4 Creating a Basic Workflow

Create a basic workflow to define the Approver and Cc role.

### Procedure

1. Go to ► *Admin Center* ► *Manage Organization, Pay and Job Structures* ►.
2. Select *Workflow* form the *Create New* dropdown.
3. Enter the *Workflow ID*, *Approve Type*, and *Approver Role*.

With this setting, you will get a To Do notification on the Home page whenever the rule is triggered.

4. In the *Cc Role* section, enter the *Contributor Type* and *Actor*.

With this setting, you will get an e-mail notification whenever the rule is triggered.

5. Click *Save*.

**Task overview:** [Configure MDF Alerts \[page 194\]](#)

## Related Information

[Creating an Object Definition \[page 195\]](#)

[Creating an Alert Message \[page 195\]](#)

[Creating an Instance for Multiple Alerts \[page 196\]](#)

[Configuring a Business Rule \[page 198\]](#)



## 13.1.5 Configuring a Business Rule

The business rule basically consists of information that triggers the alert event.

### Procedure

1. Go to ► [Admin Center](#) ► [Configure Business Rules](#) ►.
2. Click [Create New Rule](#) and select [Rules for MDF Based Objects](#) from [Metadata Framework](#) section.
3. Enter the [Rule Name](#), [Rule ID](#), [Start Date](#), [Base Object](#), and [Purpose](#) of the rule. The base object is the one you created previously in [Configure Object Definitions](#), for example, `cust_Alert`.

#### → Tip

If you want to generate alerts for each composite child record, create separate rules using the composite child object as the base object. This way, an alert is generated for each record created for the child objects.

4. Click [Continue](#).
5. Set [Always True](#) for [If](#) condition.
6. For [Then](#) condition, select [Execute](#) and then select [Trigger MDF Alert Event\(\)](#) from the dropdown.
7. Enter the values for each of the fields and save the object.

Following are the sample values for Trigger MDF Alert Event:

| Field Name           | Configuration   |
|----------------------|---|
| Workflow Information | Select <a href="#">Value</a> and then select the Workflow that you have created.              |
| Alert Due Date       | Select <a href="#">Value</a> and then select the date on which the alert should be triggered. |
| Alert Message        | Select <a href="#">Value</a> and then select the alert message that you have created.         |
| Generic Object       | Select the Generic Object for this Alert.   |

| Field Name     | Configuration  |
|----------------|--|
| MDF Alert Type | Select <i>Value</i> and then select the alert type that you have created for generating multiple alerts. |

**Note**

Remember to choose an *MDF Alert Type* if you are configuring multiple alerts for a single MDF object.

8. Click *Save*.

**Task overview:** [Configure MDF Alerts \[page 194\]](#)

## Related Information

[Creating an Object Definition \[page 195\]](#)

[Creating an Alert Message \[page 195\]](#)

[Creating an Instance for Multiple Alerts \[page 196\]](#)

[Creating a Basic Workflow \[page 197\]](#)

[Configuring Post Save Rules for an Object \[page 199\]](#)

[Creating Records for the Object \[page 200\]](#)

## 13.1.6 Configuring Post Save Rules for an Object

To trigger the alert event so that the user gets the notification about any business event, it's required to post save the rule in the *Configure Object Definitions* page.

### Procedure

1. Go to **Admin Center** > *Configure Object Definitions*.
2. Select the object that you've already created, for example cust\_Alert.
3. Click *Take Action* > *Make Correction*.
4. In the *Rules* section, select the newly created rule for *Post Save Rules* and click *Save*.

**Task overview:** [Configure MDF Alerts \[page 194\]](#)

## Related Information

- [Creating an Object Definition \[page 195\]](#)
- [Creating an Alert Message \[page 195\]](#)
- [Creating an Instance for Multiple Alerts \[page 196\]](#)
- [Creating a Basic Workflow \[page 197\]](#)
- [Configuring a Business Rule \[page 198\]](#)
- [Creating Records for the Object \[page 200\]](#)

### 13.1.7 Creating Records for the Object

After you create and save the object, you should create records for the object that is already created in the [Configure Object Definition](#) page.

#### Procedure

1. Go to [Admin Center](#) > [Manage Data](#).
2. Select the object that you have created from the [Create New](#) dropdown. For example, cust\_Alert.
3. Enter the *externalCode* to include the user. For example, Admin.
4. Enter the *effectiveStartDate* and click [Save](#).

#### Results

As soon as the rule is executed, MDF checks for the alert due date. If the due date is the current date or some date in the past, **ProcessECAAlertBySingleElement** triggers the alerts. For the future-dated alerts, it is required to run an [EC Alerts and Notification](#) jobs that is available in Provisioning.

##### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Based on the effective date (due date), the user gets an alert on the To Do list in the [Admin Center](#). The subject is displayed in the To Do list and once you expand the list, you can view the description of the alert.

**Task overview:** [Configure MDF Alerts \[page 194\]](#)

## Related Information

[Creating an Object Definition \[page 195\]](#)

[Creating an Alert Message \[page 195\]](#)

[Creating an Instance for Multiple Alerts \[page 196\]](#)

[Creating a Basic Workflow \[page 197\]](#)

[Configuring a Business Rule \[page 198\]](#)

[Configuring Post Save Rules for an Object \[page 199\]](#)

# 14 Viewing Admin Alerts for MDF Object Data Volumes

Generate admin alerts for MDF objects when their usage has exceeded the recommended limits.

## Prerequisites

To check the data volumes of an MDF object, ensure you have the relevant permissions enabled. For more information, refer to [Viewing MDF Data Volumes](#) in the Related Information.

For Data Volume Admin Alerts, ensure you have the *View*, *Edit*, *Import*, and *Export* permissions for the following:

- ▶ [User Permissions](#) ▶ [Miscellaneous Permissions](#) ▶ [MDF Association Records Alerts](#) ▶
- ▶ [User Permissions](#) ▶ [Miscellaneous Permissions](#) ▶ [MDF Audit Records Alerts](#) ▶
- ▶ [User Permissions](#) ▶ [Miscellaneous Permissions](#) ▶ [MDF Time Slice Alerts](#) ▶
- ▶ [User Permissions](#) ▶ [Miscellaneous Permissions](#) ▶ [MDF Transactional Records Alerts](#) ▶

## Context

Check data consumption of your instance against the MDF Data Volume alerts that are generated on a monthly basis, and purge the data that you no longer need.

### Note

Admin alerts for exceeding data volumes are not generated in real time, that is, these alerts are generated on a monthly basis based on the data collection schedule. The data collection schedule cannot be configured manually.

## Procedure

- Go to ▶ [Admin Center](#) ▶ [Admin Alerts](#) ▶
- In the *Admin Alert Type* dropdown, choose the alert type under *Metadata Framework (MDF)*.

The following alerts are available for MDF Data Volume:

- [MDF Data Volume - Picklist Values](#)
- [MDF Data Volume - Picklist Versions](#)
- [MDF Data Volume - Effective-dated Transactional Data](#)

- [MDF Data Volume - Audit Data](#)
- [MDF Data Volume - Total Transactional Data](#)

#### ⓘ Note

Ensure you have the necessary permissions enabled by your administrator to view these alert types.

Information on the selected alert type is displayed.

3. Choose [Subscribe to Emails](#) to set up email notifications.

We recommend that you choose [Monthly](#) subscription to these notifications as these admin alerts are generated in monthly intervals.

## Next Steps

The volume of MDF data can impact how your system perform. Based on the alerts, we recommend that you regularly monitor and purge the data that you no longer need.

#### ⓘ Note

Even after purging data that you no longer need, the respective admin alert remains in the alerts page until the next data collection has occurred and new alerts are generated.

For more information about usage limits and purging data, see the Related Information section.

## Related Information

[Viewing MDF Object Data Volumes \[page 78\]](#)

[Important Configuration Guidelines and Usage Limits \[page 10\]](#)

[Purging MDF Data \(Non-DRTM\) \[page 206\]](#)

# 15 Reporting

## 15.1 Overview

This section deals with generating reports.

For data in your client's MDF instance, reports are generated using SAP SuccessFactors Workforce Analytics. To create custom reports for your client's MDF data, you can use the report designer.

## 15.2 Prerequisites

This sections explains what is needed to set up reporting for the MDF.

For MDF data to be available in the report designer, you need to set up the correct object and field level permission security with role-based permissions (RBP).

Your access to tables and their related fields in reporting depend on their security settings as well.

### Note

You can use Advanced Reporting to report the data involving delivered and custom MDF objects, where the MDF objects can be used as a data source. Advanced Reporting requires Employee Central. If you don't have Employee Central, you can export data from MDF objects with Integration Center or MDF Import/Export.

## Related Information

[Adding Security](#)

[Advanced Reporting in the Online Report Designer](#)

## 15.3 Viewing MDF Objects from Analytics

This section explains how to access MDF objects to generate reports.

### Context

After you have set up the correct permissions, you can access MDF objects to generate reports.

### Procedure

1. In the *Home* menu, choose *Analytics*.
2. Choose the *Reporting* tab.
3. Choose ► *Tools* ► *Detailed Reporting* ▾.

If MDF is installed and the permissions are set up correctly, you see *Metadata Framework* in your list of available tables. These MDF tables are treated just like any other tables by Workforce Analytics. For information about generating reports from these tables, see [Implementing Employee Central Core](#) from the SAP Help Portal.



# 16 Purging MDF Data (Non-DRTM)

If you need to free up memory space, you can permanently remove MDF data that you no longer need. Please note that you should only use legacy and non-DRTM purge requests to purge MDF data if you've confirmed that it meets your organization's data protection and privacy requirements.

## Prerequisites

- You understand the limitations of a legacy and non-DRTM purge request, which does **not** use data retention time management (DRTM).
- You're familiar with how the Data Retention Management tool works, and also with the end-to-end-data purge process.
- You have *Create Legacy Data Purge Request* permission.

### ⚠ Caution

The system does not check if the user performing the MDF purges (non-DRTM) has permission at object level or object field level. Therefore, you should only grant this permission to users who are allowed to access all objects and object fields affected by these MDF purges.

## Context

Use these purge types to free up memory space, not for data protection and privacy. To thoroughly purge personal data, whether it's stored in MDF or not, use Data Retention Time Management (DRTM) instead.

You want to do one of the following:

- *Purge Non-Sensitive MDF Audit Data*  
This purge request type permanently removes audit data that has been created for MDF objects that have no legislatively sensitive personal data configuration. You can create the purge for both custom and predelivered MDF objects.

### ⚠ Caution

After audit data is purged, you won't be able to generate audit reports for the purged records.

### 📌 Note

In the context of an MDF audit data purge, "audit data" refers to MDF objects that have MDF Version History switched on to enable audit-logging.

- *Purge Non-Sensitive MDF Business Data*  
This purge request type permanently removes business data for MDF objects that have no legislatively sensitive personal data configuration. The purge can be run for custom and predelivered MDF objects.

- [Purge MDF Attachments](#)

This purge request type permanently removes attachments for custom and predelivered MDF objects, both **with and without** legislatively sensitive personal data configuration.

#### Note

Do not purge attachments from MDF objects with legislatively sensitive personal data unless you've confirmed that they're no longer required by your data retention requirements.

#### Note

Since attachment fields can sometimes be mandatory, the purge replaces the attachment with a file of smaller size containing a note that the attachment has been purged. This prevents any errors that might have otherwise arisen due to a missing mandatory field entry.

## Procedure

1. Go to ► [Admin Center](#) ► [Data Retention Management](#) ► [Create New Purge Request](#) ►.
2. Create a purge request:
  - To purge MDF audit data, select [Purge Non-Sensitive MDF Audit Data](#).
  - To purge MDF business data, select [Purge Non-Sensitive MDF Business Data](#).
  - To purge MDF attachments, select [Purge MDF Attachments](#).
3. Select an MDF object.

#### Note

You can't purge data for MDF object definitions and MDF picklists, nor DRTM objects.

4. Select at least one field, an operator, and a value. If you select more than one field for an object, you get only those records that meet ALL criteria.
  - For [Purge Non-Sensitive MDF Audit Data](#), select [auditTransactionDate](#) as field.

#### Caution

- If you select fields other than [auditTransactionDate](#), this could lead to gaps in the audit history and thus incorrect audit reports.
  - We don't recommend selecting translatable fields as field criteria, as this could purge more audit records than intended. A translatable field always refers to an object. For example, when the translatable field for object "1234" is purged, the system also purges any other records of object "1234" in that audit log.
5. For [Purge MDF Attachments](#), this step is optional. You can restrict the object instances for which you want to purge attachments by selecting one or more object fields with operators and values here.
5. For [Purge MDF Attachments](#), select an attachment:
    - To delete only attachments that meet specific criteria, enter size in MB and/or the last upload date as selection criteria.

- To purge **all** attachments of an attachment field, select the attachment field and then enter one of the following:
  - *Size in MB*:  $\geq 0$
  - *Last Upload Date*:  $\leq$  (select the date on which you create the purge request)

#### 📌 Note

You can select more than one attachment field for an object.

6. To make sure that MDF business data with the status *History Pending* is included in the purge, select *Include Pending History Records*.
7. Proceed with purge set up and approval, as you would for any purge request.

# 17 MDF OData API

This section provides information about MDF OData API entities including MDF Generic Object entities and MDF Foundation Object entities.

When you create a new MDF generic object (GO), you can choose whether you want to expose it to OData API. Once exposed, you can access the object through OData API calls. In this section, we'll introduce what you need to do to expose an MDF object, how to query and edit data through API calls, and what rules you need to follow when accessing MDF OData APIs.

In addition to GO entities, you'll also learn how MDF Foundation Object entities work. MDF Foundation Objects are Employee Central Foundation Objects which have been migrated to MDF and exposed to OData as OData API entities. Many MDF Foundation Objects adopt the same querying and editing behaviors as MDF Generic Objects. For more information, see [MDF Foundation Objects](#).

Before you proceed, make sure you have the basic knowledge of OData API and you're already familiar with configuring an MDF object definition:

- [OData Operations](#)
- [Configuring the Object Definition](#)

## 17.1 Exposing Custom MDF Objects to OData API

You can expose a custom MDF object to OData API by setting the *API Visibility* field when you create or edit the object definition.

### Prerequisites

MDF is enabled for your company and you have the corresponding permissions to configure object definitions. For more information, see the *Enabling MDF* section of the *Implementing the Metadata Framework (MDF)* guide.

### Context

The OData API exposes the object definition, field definition, rules, and conditions of an MDF object as OData metadata. This makes it easier for other modules to consume the data.

MDF entities that are associated with more than one parent will be exposed to OData API if one of the parents is exposed. For example, the Job Profile Builder entity makes its child objects exposed as API entities, such as job codes and competencies.

You can use the *API Visibility* field to control whether a custom MDF object is to be exposed to OData API or not, and whether it's editable or read only through API.

## Procedure

1. Go to the *Configure Object Definitions* page and choose one of the following options:
  - To create a new object definition, select *Object Definition* in the *Create New* dropdown list.
  - To edit an existing custom object definition, select *Object Definition* in the *Search* dropdown list and type the name of the object definition to search. Click and open the object definition from the result list, and choose **Take Action > Make Correction**.
2. On the *Object Definition* screen, you have the following options in the *API Visibility* field. Choose either *Editable* or *Read Only* to allow OData API access to the object:
  - *Editable* - This option allows you to create and edit object instances through OData API.
  - *Read Only* - This option only allows you to query object instances through OData API.
  - *Not Visible* - The object isn't exposed to OData API.
3. In the *API Sub Version* field, choose the subversion for your object. For more information, see [API Subversioning \[page 211\]](#).
4. Proceed with the rest of the configuration tasks and choose *Save*.

## Results

The custom MDF object is now exposed to OData. The OData metadata will be automatically refreshed shortly after and you're able to access the object through the corresponding OData API.

### Note

Depending on the frequency of metadata refreshing job scheduled for your instance, the change may not take effect immediately. You can manually refresh the metadata using one of the following approaches:

- Go to **Admin Center > OData Metadata Refresh and Export** and choose *Refresh* to refresh the metadata.
- Send an API request to refresh the metadata:

```
GET https://<API-endpoint-URL>/odata/v2/refreshMetadata
```

## 17.1.1 API Subversioning

You can control whether you want to expose system fields of an MDF object to the OData API with API subversioning.

An MDF object contains a list of technical system fields. When exposed to OData API, all of them may not be needed. With the API subversioning feature, you can choose to hide these fields in the MDF OData API entity.

Depending on the *MDF system field in API Mode* Provisioning settings for your company, the system fields may be:

- Not exposed for all MDF entities regardless of the *API Sub Version* field setting in the MDF object definition
- Exposed for all MDF entities regardless of the *API Sub Version* field setting in the MDF object definition
- (By default) Dependent on the *API Sub Version* field setting of the MDF object definition

### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

In *Configure Object Definitions*, you have the following options in the *API Sub Version* field:

- V1.0: System fields following the rules of the subversion are exposed.
- V1.1: System fields following the rules of the subversion are hidden.
- No Selection: The same as using V1.1.

### ⓘ Note

The *API Sub Version* setting takes effect only when the Provisioning setting of MDF system field for API is set to be dependent on it. Otherwise, this field is ignored.

Selecting or changing to an API subversion can affect the metadata of the OData API. As a result, your API request may fail because a nonexisting property. We recommend that you check the metadata of the MDF OData API entity and compare it against your API request as a necessary troubleshooting step.

## List of MDF System Fields

The following table lists all MDF system fields and whether they can be removed:

| Field Name              | Following the rule of subversions | Notes  |
|-------------------------|-----------------------------------|--|
| mdfSystemCreatedBy      | Yes                               | Equivalent to <code>createdBy</code> property in OData       |
| mdfSystemCreatedDate    | Yes                               | Equivalent to <code>createdDateTime</code> property in OData |
| mdfSystemLastModifiedby | Yes                               | Equivalent to <code>lastModifiedBy</code> property in OData  |

| Field Name  | Following the rule of subversions | Notes   |
|---|-----------------------------------|---|
| mdfSystemLastModifiedDate   | Yes                               | Equivalent to <code>LastModifiedDateTime</code> property in OData                                     |
| mdfSystemLastModifiedDateWithTZ   | Yes                               | Equivalent to <code>LastModifiedDateTime</code> property in OData                                     |
| LastModifiedDateWithTZ  | Yes                               | Equivalent to <code>LastModifiedDateTime</code> property in OData                                     |
| mdfSystemEntityId   | Depends                           | Exposed when the field name doesn't start with <code>mdfSystem</code>                                 |
| mdfSystemRecordId   | Depends                           |   |
| <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>Note</b></p> <p>Users can overwrite the system field name during object definition. If the user-defined name doesn't start with <code>mdfSystem</code>, this field is exposed.</p> </div> |                                   |   |
| mdfSystemObjectType   | Yes                               |   |
| mdfSystemStatus   | Depends                           | Exposed when the field visibility is set to be visible  |
| mdfSystemStatusNav  | Depends                           | Exposed when the field visibility is set to be visible  |
| mdfSystemRecordStatus   | No                                | This field indicates pending data and won't be removed in OData API                                   |
| mdfSystemVersionId  | Yes                               | This field indicates a pending history record. Pending history records aren't supported in OData API. |
| mdfSystemEffectiveStartDate   | Yes                               | This field can be removed when the MDF object isn't effective dated.                                  |
| mdfSystemEffectiveEndDate   | Yes                               | This field can be removed when the MDF object isn't effective dated.                                  |
| mdfSystemTransactionSequence  | Yes                               | This field can be removed when the effective dating of the MDF object isn't Multiple Changes Per Day. |

## 17.2 MDF Foundation Object Entities

Employee Central Foundation Objects migrated to the Metadata Framework (MDF).

MDF Foundation Object entities are Employee Central Foundation Objects that have been migrated to the Metadata Framework (MDF) and exposed as OData API entities. MDF Foundation Object entities follow the general rules of MDF OData operations with a few exceptions.

Here's a list of MDF Foundation Object entities:

| OData API Entity                   | MDF Foundation Object               |
|------------------------------------|-------------------------------------|
| FOBusinessUnit                     | Business Unit                       |
| FOCompany                          | Legal Entity                        |
| FOCostCenter                       | Cost Center                         |
| FODepartment                       | Department                          |
| FODivision                         | Division                            |
| FOJobCode                          | Job Classification                  |
| FOJobClassLocal<Country/Region>    | Job Classification <Country/Region> |
| FOJobFunction                      | Job Function                        |
| FOLegalEntityLocal<Country/Region> | Legal Entity <Country/Region>       |
| FOPayGroup                         | Pay Group                           |

For more information, see [MDF Foundation Objects](#).

## 17.3 Data Mapping Between MDF and OData API

This topic contains information about the mapping of MDF objects to the OData API entities.

Each MDF object can be mapped to one OData entity only. Mapping an MDF object to multiple OData entities is not allowed. When an MDF object is exposed to OData API, the code of the MDF object is used as the name of the OData API entity type. In the following sections, you'll find detailed mapping relationships between an MDF object and its corresponding OData API entity.

### Note

Composite objects with multiple parents will not be exposed to OData API.

If a parent object is not exposed to OData API, none of its composite children will be exposed regardless of their API visibility settings.

## Mapping of General Terms

| MDF Object Definition | OData API                    |
|-----------------------|------------------------------|
| Object                | Entity Set or Entity Type    |
| Field                 | Property/Navigation Property |
| Association           | Navigation Property          |



## Mapping of Attributes

The following table shows the mapping relationship between the MDF field/association attributes and OData entity set attributes.

| Property Attribute of an OData Entity Type | Attribute of the Corresponding MDF Field/Association | Mapping Rule   |
|--|--|--|
| Property name                              | Field/association name                               | <p>The following rules apply:</p> <ul style="list-style-type: none"> <li>In general, the name of OData property attribute is the same as the attribute name of the MDF object.</li> <li>If an MDF field is linked to another object, or if it's an association, the name appears in OData with the 'Nav' suffix. See the <a href="#">Mapping of Data Types [page 216]</a> section for more details.</li> <li>The following MDF field names are hardcoded and appear "as-is" in OData: <ul style="list-style-type: none"> <li>createdBy</li> <li>createdDateTime</li> <li>lastModifiedBy</li> <li>lastModifiedDateTime</li> </ul> </li> </ul> |
| Property type                              | Field data type                                      | For detailed mapping of data types, see <a href="#">Mapping of Data Types [page 216]</a> .   |
| Nullable                                   | There is no equivalent attribute in MDF.             | The <code>sap:nullable</code> property is always true.   |
| Required                                   | Required and Visibility                              | <p>When the API Visibility of the MDF object is set to Read Only, the values of <code>sap:required</code>, <code>sap:creatable</code>, <code>sap:updatable</code>, and <code>sap:upsertable</code> attributes are always false regardless of the field required and visibility settings.</p> <p>When the API Visibility of the MDF object is set to Editable, these field values correspond to the <i>Required</i> and <i>Visibility</i> attributes of the MDF fields.</p> <p>For a detailed mapping of these attributes, see the table below in <a href="#">Mapping of Edit Attributes</a>.</p>   |
| Creatable                                  | Visibility   |  |
| Updatable                                  | Visibility   |  |
| Upsertable                                 | Visibility   |  |
| Visible                                    | There is no equivalent attribute in MDF.             | The <code>sap:visible</code> attribute is always true.   |
| Sortable                                   | There is no equivalent attribute in MDF.             | The <code>sap:sortable</code> attribute is always true except for the CLOB-type field and localized property of a translatable field.  |
| Filterable                                 | There is no equivalent attribute in MDF.             | The <code>sap:filterable</code> attribute is always true except for the CLOB-type field and localized property of a translatable field.  |

## Mapping of Edit Attributes

The following table explains how the visibility and edit attributes of an MDF field works when exposed to OData API.

### ⓘ Note

When a MDF field maps to both property and navigation property in OData, the *sap:required* attribute of these properties may take different values. These exception are documented in the notes below.

| MDF Field Attribute |            | OData Property Attribute   |               |               |                |
|---------------------|------------|--|---------------|---------------|----------------|
| Required            | Visibility | sap:required   | sap:creatable | sap:updatable | sap:upsertable |
| Yes                 | Editable   | true   | true          | true          | true           |
|                     |            | <h3>ⓘ Note</h3> <p>Exception: for properties mapped from Generic Objects, User, and Picklist, and navigation properties mapped from Enum type, the value is false.</p>                 |               |               |                |
| Yes                 | Read Only  | false  | false         | false         | false          |
|                     |            | <h3>ⓘ Note</h3> <p>Exception: this value is true for navigation properties that are mapped from associations and from these types: Attachment, Generic Object, Picklist, and User.</p> |               |               |                |

| MDF Field Attribute |             | OData Property Attribute  |               |               |                |
|---------------------|-------------|---|---------------|---------------|----------------|
| Required            | Visibility  | sap:required  | sap:creatable | sap:updatable | sap:upsertable |
| Yes                 | Not Visible | false   | false         | false         | false          |
|                     |             | <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p><b>Note</b></p> <p>Exception: this value is true for navigation properties that are mapped from associations and from these types: Attachment, Generic Object, Picklist, and User.</p> </div> |               |               |                |
| No                  | Editable    | false   | true          | true          | true           |
| No                  | Read Only   | false   | false         | false         | false          |
| No                  | Not Visible | false   | false         | false         | false          |

For transient fields and optimistic lock UUID, the following rules apply:

|  | sap:required | sap:creatable | sap:updatable | sap:upsertable | sap:sortable | sap:filterable |
|--|--------------|---------------|---------------|----------------|--------------|----------------|
| Transient fields<br>( <a href="#">What is a transient field?</a> ) | false        | false         | false         | false          | false        | false          |
| Optimistic Lock UUID   | false        | false         | true          | true           | false        | false          |

## Mapping of Data Types

The following table shows the mapping relationship between the data types of MDF fields and OData properties.

| MDF Field Type | OData Property Type  | Naming Rule   | Max Length | Multiplicity (For Navigation Properties) |
|----------------|--|---|------------|--|
| Attachment     | Navigation property to the Attachment entity.  | Add Nav suffix to MDF field name. Example: cust_MyAttachmentNav | N/A        | *:1                                      |
|                | <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>Note</b></p> <p>Inline editing of the attachment navigation property is not allowed. To add an attachment for an MDF entity, you first create the attachment with the Attachment OData API, and then add the attachment to the MDF entity. For more information, see <a href="#">Linking Attachments to an MDF Entity [page 268]</a>.</p> </div> |   |            |  |
| Auto Complete  | String   | Same name as MDF field.   | N/A        | N/A                                      |
| Auto Number    | Int64  | Same name as MDF field.   | N/A        | N/A                                      |
| Boolean        | Boolean  | Same name as MDF field.   | N/A        | N/A                                      |
| CLOB           | String   | Same name as MDF field.   | N/A        | N/A                                      |
|                | <p>You can use this data type to store a block of text. The advantage of this data type is that access is fast because an application program can access any portion of a CLOB object. Additionally, this data type provides large storage for a user-defined data type.</p>   |   |            |  |
| Date           | DateTime   | Same name as MDF field.   | N/A        | N/A                                      |

| MDF Field Type    | OData Property Type  | Naming Rule   | Max Length   | Multiplicity (For Navigation Properties)  |
|-------------------|--|---|--|---|
| DateTime          | DateTimeOffset   | Same name with the following exceptions: <ul style="list-style-type: none"> <li>MDF field <code>createdDate</code> maps to OData property type <code>createdDateTime</code></li> <li>MDF field <code>lastModifiedDate</code> maps to OData property type <code>lastModifiedDateTime</code></li> </ul> | N/A  | N/A   |
| Decimal           | Decimal  | Same name as MDF field.   | N/A  | N/A   |
| Enum              | One string property.<br>One navigation property to the <code>MDFEnumValue</code> entity. | Property: same name as MDF field name.<br>Navigation property: add <code>Nav</code> suffix to MDF field name. Example: <code>cust_MyEnumNav</code>  | Same max length as defined in MDF field. Default value is 255. | *:1   |
| Foundation Object | One string property.<br>One navigation property to the Foundation Object entity.         | Property: same name as MDF field.<br>Navigation property: add <code>Nav</code> suffix to MDF field name. Example: <code>cust_MyObjectNav</code>   | 32   | *:1   |
| Generic Object    | One string property.<br>One navigation property to the Generic Object entity.            | Property: same name as MDF field.<br>Navigation property: add <code>Nav</code> suffix to MDF field name. Example: <code>cust_MyObjectNav</code>   | 128  | If the source object is not effective dated and destination object is effective dated, or if the destination object is <code>PickListValue</code> the multiplicity is <code>*:*</code> .<br><br>Otherwise, if the field is required, the multiplicity is <code>*:1</code> ; if it's optional, the multiplicity is <code>*:0..1</code> . |

| MDF Field Type | OData Property Type   | Naming Rule  | Max Length | Multiplicity (For Navigation Properties)   |
|----------------|---|--|------------|--|
| Number         | Int64   | Same name as MDF field.  | N/A        | N/A  |
| Picklist       | One string property.<br>One navigation property to the destination picklist entity.   | Property: same name as MDF field name.<br>Navigation property: add <code>Nav</code> suffix to MDF field name. Example: <code>cust_MyPicklistNav</code> | N/A        | If the source object is effective dated, the multiplicity is <code>*:*</code> .<br><br>Otherwise, if the field is required, the multiplicity is <code>*:1</code> ; if it's optional, the multiplicity is <code>*:0..1</code> . |
| String         | String  | Same name as MDF field.  | N/A        | N/A  |
| Time           | Time<br><br>On UI, times are entered and displayed in <b>HH:MM:SS</b> format, for example: <b>10:30:59</b> .<br>In OData API, times are entered and displayed in duration format, for example: <b>PT10H30M59S</b> | Same name as MDF field.  | N/A        | N/A  |

| MDF Field Type | OData Property Type  | Naming Rule  | Max Length   | Multiplicity (For Navigation Properties) |
|----------------|--|--|--|--|
| Translatable   | <p>One string property for default value, one string property for localized label, and a set of string properties representing labels in each locale with locale names added as suffixes.</p> <p>One navigation property to the <code>MDFLocalizedValue</code> entity.</p> | <p>Default value property: <code>add_defaultValue</code> suffix to MDF field name. Example, <code>cust_myLabel_defaultValue</code>.</p> <p>Localized value property: <code>add_localized</code> suffix to MDF field name. Example, <code>cust_myLabel_localized</code>.</p> <p>Locale properties: add locale suffixes to MDF field names. Examples: <code>cust_myLabel_en_US</code>, <code>cust_myLabel_zh_CN</code>.</p> <p>Navigation property: <code>add_translationTextNav</code> suffix to the MDF field name. Example: <code>cust_myLabel_translationTextNav</code>.</p> | Same max length as defined in MDF field. Default value is 255. | 1:*                                      |

| MDF Field Type | OData Property Type  | Naming Rule  | Max Length | Multiplicity (For Navigation Properties)               |
|----------------|--|--|------------|--|
| User           | One string property.<br>One navigation property to <code>User</code> entity. | Property: same name as MDF field name.<br>Navigation property: add <code>Nav</code> suffix to MDF field name. Example: <code>cust_MyUserNav</code> | 100        | 1:1 if the field is required; 1:0..1 if it's optional. |

**Note**

The user-type audit fields `mdfSystemCreatedBy` and `mdfSystemLastModifiedBy` are mapped to the following read-only properties in OData API by default:

- Properties: `createdBy`, `lastModifiedBy`
- Navigation Properties: `createdByNav`, `lastModifiedByNav`

Reverse navigation is not allowed for the navigation properties.



| MDF Field Type | OData Property Type | Naming Rule   | Max Length | Multiplicity (For Navigation Properties) |
|----------------|---------------------|---|------------|--|
| -              | wfRequestNav        | When workflow-enabled MDF objects are exposed to OData, a wfRequestNav navigation property is added automatically to the OData API entity. You can use this navigation property to retrieve the related workflow information for pending data.<br><br>For more information, see <a href="#">Data Mapping Between MDF and OData API [page 213]</a> . | N/A        | 1:0..*                                   |

## Mapping of Associations

The associations in an MDF object are mapped to navigation properties in OData API with the same names. The associations are mapped as shown in the following table.

| MDF Association | Mapped to OData API   | Multiplicity   |
|-----------------|---|--|
| Composite       | A navigation property to the <b>child</b> MDF entity in the association.    | If it's a 1:M association, or if the source object is not effective dated and destination object is effective dated, the multiplicity is 1:*.<br><br>Otherwise, if the association is required, the multiplicity is 1:1; if it's optional, the multiplicity is 1:0..1. |
| Valid When      | A navigation property to the <b>referred</b> MDF entity in the association. | If it's a 1:M association, or if the source object is not effective dated and destination object is effective dated, the multiplicity is *:*.<br><br>Otherwise, if the association is required, the multiplicity is *:1; if it's optional, the multiplicity is *:0..1. |

| MDF Association | Mapped to OData API  | Multiplicity   |
|-----------------|--|--|
| Join By Column  | A navigation property to the <b>referred</b> destination entity. | <p>If it's a 1:M association, or if the source object is not effective dated and destination object is effective dated, the multiplicity is *:*</p> <p>Otherwise, if the association is required, the multiplicity is *:1; if it's optional, the multiplicity is *:0..1.</p> |

## Defining Key Properties for OData Entity

The key properties of an MDF OData API entity depend on the effective dating setting for the MDF object. The following table explains which MDF fields are used as keys in OData.

### Note

The field names referred to in this table reflect the default names which correspond to the Database Field Name column. If you have changed the default names, the key property names should reflect these changes.

| Effective Dating         | Number of Keys | Key Properties  |
|--------------------------|----------------|---|
| None                     | 1              | externalCode  |
| Basic                    | 2              | externalCode<br>effectiveStartDate                        |
| Multiple Changes Per Day | 3              | externalCode<br>effectiveStartDate<br>transactionSequence |

| Effective Dating | Number of Keys                            | Key Properties   |
|------------------|---|--|
| From Parent      | 2 or more depending on the parent entity. | <p><code>externalCode</code></p> <p><code>&lt;ParentEntityName&gt;_externalCode</code></p> <p><code>&lt;ParentEntityName&gt;_effectiveStartDate</code> - Only available if the effective dating of parent entity is Basic or Multiple Changes Per Day</p> <p><code>&lt;ParentEntityName&gt;_transactionSequence</code> - Only available if the effective dating of parent entity is Multiple Changes Per Day</p> |

**Note**

In a multiple-level parent-child relation, the child inherits the `<ParentEntityName>_externalCode` key properties from all its ancestors. That means if there's a 3-level relationship, the "youngest" child will have two `<ParentEntityName>_externalCode` key properties in addition to its own `externalCode`, one from its parent, and the other from the parent's parent. However, there will be only one `<ParentEntityName>_effectiveStartDate` as the effective start date is the same for all entities in a composite association.

## 17.4 Locale Handling in MDF Entities

How MDF handles translatable text fields in OData API.

MDF exposes translatable text fields as OData properties. For each translatable field, these standard properties and navigation properties are available in the entity metadata:

- Property `<fieldname_defaultValue>` represents the default property value. You can assign a value to the property when you create an entry. Otherwise, it takes the value of the locale-specific property.
- Property `<fieldname_localized>`: represents the localized property value in the current locale. The localized value calculated following the rules described in [How Is the Localized Value Determined? \[page 225\]](#).
- Property `<fieldname_locale code>`: represents the localized value in the locale indicated by the locale code.
- Navigation property `<fieldnameTranslationTextNav>`: represents the association with entity `MDFLocalizedValue`, where the localized values of the properties are stored.

For example, a description field in an MDF object has a default property `<description_defaultValue>`, a localized property `<description_localized>`, several locale-specific properties such as `<description_en_US>`, `<description_de_DE>`, `<description_zh_CN>`, etc., and a navigation property `<descriptionTranslationTextNav>`.

All these properties and navigation properties can be queried and edited.

## How Is the Localized Value Determined?

The system determines the value of the `<fieldname_localized>` field based on the following information:

- User locale. The user local corresponds to the language setting the global header bar under [Settings](#) [Languages](#).
- Company locale. The company locale depends on the language setting of your company in Provisioning.

### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

- Value of the `<fieldname_defaultValue>` property.

Here's how the `<fieldname_localized>` property value is determined:

1. If the user's locale is available, it takes the property value that corresponds to the user's locale.
2. If user's locale isn't available, it takes the property value that corresponds to company's locale.
3. If neither locale is available, it takes the property value of `<fieldname_defaultValue>`.

### ⓘ Note

Both queries and edits follow the same rules. If you edit the value of `<fieldname_localized>`, the corresponding field of the locale is also changed. For example, the `<name_localized>` field of the `FOBusinessObject` in your company comes from `en_US` field. If you modify this field with an API request, the new value is also reflected in the `<name_en_US>` field.

## Example 1: Query an Entry's Localized Label

This type of request can be useful especially for UI consumption where you need to retrieve a record with the labels of the default locale. Here's a sample request using the `FOBusinessUnit` entity as an example:

### Request

| Operation   | Query  |
|-------------|--|
| HTTP Method | GET  |
| URI         | <code>https://&lt;API-Server&gt;/odata/v2/FOBusinessUnit(externalCode='MC_AVIA',startDate=datetime'1900-01-01T00:00:00')?\$format=JSON&amp;\$select=externalCode,name_localized</code> |

### Response

```
{
  "d": {
    "__metadata": {
```

```

        "uri": "https://<API-Server>/odata/v2/
FOBusinessUnit(externalCode='MC_AVIA',startDate=datetime'1900-01-01T00:00:00')",
        "type": "SFOData.FOBusinessUnit"
    },
    "externalCode": "MC_AVIA",
    "name_localized": "MC Aviation"
}
}
}

```

## Example 2: Query All Labels of an Entry

You often find this type of usage in integration scenarios where you want to export the labels in all languages.

### Request

| Operation   | Query  |
|-------------|--|
| HTTP Method | GET  |
| URI         | <pre> https://&lt;API-Server&gt;/odata/v2/ FOBusinessUnit(externalCode='SVCS',startDa te=datetime'1900-01-01T00:00:00')/ nameTranslationTextNav?\$format=JSON </pre> |

### Response

```

{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('en_US')",
          "type": "SFOData.MDFLocalizedValue"
        },
        "locale": "en_US",
        "value": "Services"
      },
      {
        "__metadata": {
          "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('defaultValue')",
          "type": "SFOData.MDFLocalizedValue"
        },
        "locale": "defaultValue",
        "value": "Services"
      },
      {
        "__metadata": {
          "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('en_GB')",
          "type": "SFOData.MDFLocalizedValue"
        },
        "locale": "en_GB",
        "value": "Services"
      },
      {
        "__metadata": {
          "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('zh_CN')",
          "type": "SFOData.MDFLocalizedValue"
        }
      }
    ]
  }
}

```

```

    },
    "locale": "zh_CN",
    "value": "服务"
  },
  {
    "__metadata": {
      "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('pt_BR')",
      "type": "SFOData.MDFLocalizedValue"
    },
    "locale": "pt_BR",
    "value": "Serviços"
  },
  {
    "__metadata": {
      "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('ja_JP')",
      "type": "SFOData.MDFLocalizedValue"
    },
    "locale": "ja_JP",
    "value": "サービス統括本部"
  },
  {
    "__metadata": {
      "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('ru_RU')",
      "type": "SFOData.MDFLocalizedValue"
    },
    "locale": "ru_RU",
    "value": "Сервисы"
  },
  {
    "__metadata": {
      "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('es_ES')",
      "type": "SFOData.MDFLocalizedValue"
    },
    "locale": "es_ES",
    "value": "Servicios"
  },
  {
    "__metadata": {
      "uri": "https://<API-Server>/odata/v2/
MDFLocalizedValue('ko_KR')",
      "type": "SFOData.MDFLocalizedValue"
    },
    "locale": "ko_KR",
    "value": "서비스"
  }
]
}
}}

```

### Example 3: Creating or Editing an Entry with Localized Texts

When you create or edit an entry, you can include the localized texts of translatable fields either through the local-specific properties or through inline-editing the `<fieldname>TranslationTextNav` navigation property. Here are some examples:

#### Request Info

|             |   |
|-------------|---|
| Operation   | Upsert  |
| HTTP Method | POST  |
| URI         | <code>https://&lt;API-Server&gt;/odata/v2/upsert</code> |

Payload example for upserting property values:

```
{
  "__metadata": {
    "uri": "FOBusinessUnit"
  },
  "externalCode": "SERVICE",
  "status": "A",
  "startDate": "/Date(1508375121000)/",
  "name_ko_KR": "서비스",
  "name_pt_BR": "Serviços",
  "name_de_DE": "Dienstleistungen",
  "name_zh_TW": "服務",
  "name_es_ES": "Servicios",
  "name_ru_RU": "Сервисы",
  "name_fr_FR": "Services",
  "name_ja_JP": "サービス統括本部",
  "name_en_US": "Services",
  "name_zh_CN": "服务"
}
```

Payload example for inline-upserting the navigation property:

#### Sample Code

```
{
  "__metadata": {
    "uri": "FOBusinessUnit"
  },
  "externalCode": "SERVICE",
  "status": "A",
  "startDate": "/Date(1508375121000)/",
  "nameTranslationTextNav": [
    {
      "locale": "ko_KR",
      "value": "서비스"
    },
    {
      "locale": "pt_BR",
      "value": "Serviços"
    },
    {
      "locale": "de_DE",
      "value": "Dienstleistungen"
    },
    {
      "locale": "zh_TW",
      "value": "服務"
    }
  ]
}
```

```

    {
      "locale": "es_ES",
      "value": "Servicios"
    },
    {
      "locale": "ru_RU",
      "value": "Сервисы"
    },
    {
      "locale": "fr_FR",
      "value": "Services"
    },
    {
      "locale": "ja_JP",
      "value": "サービス統括本部"
    },
    {
      "locale": "en_US",
      "value": "Services"
    },
    {
      "locale": "zh_CN",
      "value": "服务"
    }
  ]
}

```

## Response

```

{
  "d": [
    {
      "key": "FOBusinessUnit/externalCode=SERVICE,FOBusinessUnit/
startDate=2017-10-19T00:00:00.000-04:00",
      "status": "OK",
      "editStatus": "UPsertED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}

```

## 17.5 MDF OData API Operations

Describes the MDF OData API behaviors with examples.

MDF OData API is based on SAP SuccessFactors HCM suite OData API framework, currently on OData Version 2.0. While the majority of MDF OData API operations follow the same rules defined by the framework, there are patterns specific to MDF when you use MDF OData API entities. In this section, you'll find out details about the specific patterns and how they affect your OData API usage.



## Service Limit for Large MDF OData Requests

There's a service limit for large MDF OData requests. When you query or edit an MDF OData entity and the response returns a large number of records that exceed the system-defined limit, the request fails and an error occurs. You need to adjust the request parameters or payload so that the data size returned is within the limit. Depending on your request, the limit may vary.

### → Remember

The service limit is designed to optimize MDF OData performance while ensuring all MDF OData APIs continue to work in all instances. It has very little impact on your current integrations and UIs.

If you meet such an error, here are a few steps that can help solve the problem:

- Use pagination in your query if you haven't already.
- If you're already paging your query, try tuning down the page size.
- Include fewer records in your request payload.
- A rule of thumb is to cut the page or record size to half.

If you wish to change the limit, contact Technical Support.

## 17.5.1 Permissions

Permissions required for accessing MDF OData API entities.

The following table lists the required permissions for users to query and edit MDF OData API entities:

| Permission   | Description   |
|--|---|
| <a href="#">▶ Administrator Permissions</a> > <a href="#">Manage Integration Tools</a><br><a href="#">▶ Allow Admin to Access OData API through Basic Authentication</a> ▶ | This permission allows users to access OData API entities through Basic Authentication. If you're using other authentication methods such as OAuth 2.0, this permission isn't required.   |
| <a href="#">▶ Administrator Permissions</a> > <a href="#">Metadata Framework</a> > <a href="#">Admin access to MDF OData API</a> ▶   | This is the MDF OData admin permission that allows users to query and edit all MDF OData entities. It overrides all entity-level permissions.<br><br>This permission is recommended only for integration scenarios. Permission roles granted with this permission can override other MDF data access permission configurations and save data without approvals. |

| Permission  | Description   |
|---|---|
| <a href="#">▶ Administrator Permissions</a> > <a href="#">Metadata Framework</a><br><a href="#">▶ Access to non-secured objects</a> > | <p>This is the non-admin MDF OData permission. This permission allows users to query and edit all nonsecured MDF OData entities.</p> <div data-bbox="818 405 1427 554" style="background-color: #f0f0f0; padding: 10px;"> <p><b>Note</b></p> <p>This permission isn't required for querying and editing secured MDF OData entities.</p> </div>  |
| <p>Entity-level permissions for secured objects</p>   | <p>If an MDF object is secured, an entity-level permission item is available under the corresponding permission category specified in object definition. You can enable view, edit, import, and export permission for this entity. You can also maintain field-level permission overrides.</p> <p>For more information, see <a href="#">Adding Security</a> and <a href="#">Setting Up Security for Fields</a>.</p> |

## 17.5.2 Query Operation

This section contains information about query operations on MDF OData entities, including query parameters and examples. Note that only MDF-specific information is documented here. For more information on general OData API operations, refer to the OData Operations section.

[Query Parameters \[page 231\]](#)

A list of query parameters MDF OData API and usage examples.

### 17.5.2.1 Query Parameters

A list of query parameters MDF OData API and usage examples.

**Parent topic:** [Query Operation \[page 231\]](#)

## 17.5.2.1.1 asOfDate

Use the `asOfDate` parameter to query a single record that is effective on a specific date.

### Parameter Values

| Value                           | Description                                   |
|---------------------------------|---|
| <code>&lt;yyyy-mm-dd&gt;</code> | A numeric date value, for example: 2019-12-25 |

**Note**  
Returns only one record that is effective on the specified date.

**Note**  
Depending on the parameters you use, a query may return zero, one, or multiple records.  
If the effective dating type of the entity is Multiple Changes Per Day (MCPD), the record with the largest sequence number is returned as the effective record of that date.  
If you don't specify any date or date period, the query uses today's date as the `asOfDate` value and fetch the current effective record.

### Use Case

**Tip**  
When you query an effective-dated entity without any date parameters, the result returns a single record effective on the present date. It defaults to `&asOfDate=<today's date>`.

The `asOfDate` parameter retrieves the single records of an entity that is effective on the specified date.

Example:

```
https://<API-Server>/odata/v2/MDFEntity?$format=JSON&asOfDate=2018-11-07
```

### Related Information

[Using asOfDate with \\$expand \[page 238\]](#)

## 17.5.2.1.2 fromDate

Use the `fromDate` parameter to query records that are effective after a specific date.

### Parameter Values

| Value                           | Description                                   |
|---------------------------------|---|
| <code>&lt;yyyy-mm-dd&gt;</code> | A numeric date value, for example: 2019-12-25 |

**Note**  
Multiple records can be returned.

**Note**  
Depending on the parameters you use, a query may return zero, one, or multiple records.  
If the effective dating type of the entity is Multiple Changes Per Day (MCPD), the record with the largest sequence number is returned as the effective record of that date.  
If you don't specify any date or date period, the query uses today's date as the `asOfDate` value and fetch the current effective record.

### Use Case

The `fromDate` parameter retrieves all effective records of the entity in the specified time period. If the `mdfSystemEffectiveEndDate` of a record is later than the specified date, the record is returned.

Example:

```
https://<API-Server>/odata/v2/MDFEntity?$format=JSON&fromDate=2018-11-07
```

### Related Information

[Using fromDate with \\$expand \[page 240\]](#)

## 17.5.2.1.3 toDate

Use the `toDate` parameter to query records that are effective before a specific date.

### Parameter Values

| Value                           | Description                                   |
|---------------------------------|---|
| <code>&lt;yyyy-mm-dd&gt;</code> | A numeric date value, for example: 2019-12-25 |

**Note**  
Multiple records can be returned in the result.

**Note**  
Depending on the parameters you use, a query may return zero, one, or multiple records.  
If the effective dating type of the entity is Multiple Changes Per Day (MCPD), the record with the largest sequence number is returned as the effective record of that date.  
If you don't specify any date or date period, the query uses today's date as the `asOfDate` value and fetch the current effective record.

### Use Case

The `toDate` parameter retrieves all effective records of the entity before the specified date. If the `mdfSystemEffectiveStartDate` of a record is earlier than the specified date, the record is returned.

Example:

```
https://<API-Server>/odata/v2/MDFEntity?$format=JSON&toDate=2018-11-07
```

### Related Information

[Using toDate with \\$expand \[page 243\]](#)

## 17.5.2.1.4 filterParentDate

Use the `filterParentDate` parameter to query a child of an effective-dated entity with date parameters.

In a composite MDF association, a child entity inherits its effective dating attribute from its parent. When you query a composite child entity, date parameters such as `fromDate`, `toDate`, and `asOfDate` are ignored. In order

for them to work, you must include parameter `filterParentDate` with value `true` in the query. This parameter allows you to filter the child entity based on the effective dates of its parent.

## Parameter Values

| Value                        | Description   |
|------------------------------|---|
| <code>true</code>            | Filter child entity based on the effective dates of parent. |
| <code>false</code> (Default) | Date parameters are ignored on child entities.              |

## Use Case

See the topic in [Related Links](#) section for detailed use cases.

## Related Information

[Querying the Child of an Effective Dated Entity \[page 246\]](#)

## 17.5.2.1.5 recordStatus

Use the `recordStatus` parameter to query normal, pending, or pending history data of an MDF entity.

## Parameter Values

| Value                         | Description  |
|-------------------------------|--|
| <code>normal</code> (default) | Parameter value to retrieve normal records.<br><br>If <code>recordStatus</code> isn't specified, the query returns normal data by default.   |
| <code>pending</code>          | Parameter value to retrieve records pending approval.  |
| <code>pendinghistory</code>   | A record in pending history status can be: <ul style="list-style-type: none"><li>• Declined by the approver</li><li>• Canceled by the initiator of the workflow</li><li>• Modified by the approver</li></ul> |

## ⚠ Caution

To query records with pending and pendinghistory status, you must have the [Admin access to MDF OData API](#) permission.

## Use Case

In MDF OData API entities, if workflow routing is configured for an object and the [Pending Data](#) option is enabled, data changes pending approval are stored as pending data. Once the data changes are approved, the records are saved as normal data. If a workflow is declined, canceled, or modified, a pending history is stored in the database. The following table explains how the parameter values map to the `mdfSystemRecordStatus` field of the MDF object:

| Parameter Value | mdfSystemRecordStatus |
|-----------------|-----------------------|
| normal          | <b>N</b>              |
| pending         | <b>P</b>              |
| pendinghistory  | <b>PH</b>             |

You can use parameter `recordStatus` in an OData query to specify which type of data to retrieve.

- To retrieve normal records:

```
https://<API-Server>/odata/v2/cust_TestObject?$format=JSON&recordStatus=normal
```

- To retrieve pending records:

```
https://<API-Server>/odata/v2/cust_TestObject?$format=JSON&recordStatus=pending
```

- To retrieve pending history records:

```
https://<API-Server>/odata/v2/cust_TestObject?$format=JSON&recordStatus=pendinghistory
```

## 📌 Note

This parameter is required if you want to query the workflow information of an MDF entity by expanding `wfRequestNav`. For more information, see [Retrieving Workflow Information for Pending Data \[page 252\]](#).

## Error Codes

| Error Code              | Description  |
|-------------------------|--|
| COE_UNSUPPORTED_FEATURE | <p>[COE0025]Unsupported feature: Expand workflow navigation from normal data is NOT supported.</p> <p>If you try to expand wfRequestNav on a normal entry, this error occurs. wfRequestNav can only be expanded if it's a pending entry.</p> |
| COE_GENERAL_FORBIDDEN   | <p>[COE0020]Non admin user is NOT allowed to query pending data.</p> <p>Only users with the <a href="#">Admin access to MDF OData API</a> permission can query pending data.</p>   |
| COE_GENERAL_FORBIDDEN   | <p>[COE0025]Unsupported feature: Non admin user query pending history data is NOT supported.</p> <p>Only users with the <a href="#">Admin access to MDF OData API</a> permission can query pending data.</p>                                 |

### 17.5.2.1.6 versionId

Use the `versionId` parameter to query a specific version of a pending record.

### Parameter Values

| Value                          | Description  |
|--------------------------------|--|
| <largest version ID> (default) | If versionId is not specified, the query returns the record with the largest version ID. |
| <user specified version ID>    | The record with specified version ID is returned.  |

### Use Case

#### Note

If you use `versionId` in a query, it returns pending data by default.



A record pending approval can have multiple versions. You can use parameter `versionId` to query a specific version of the record. Example:

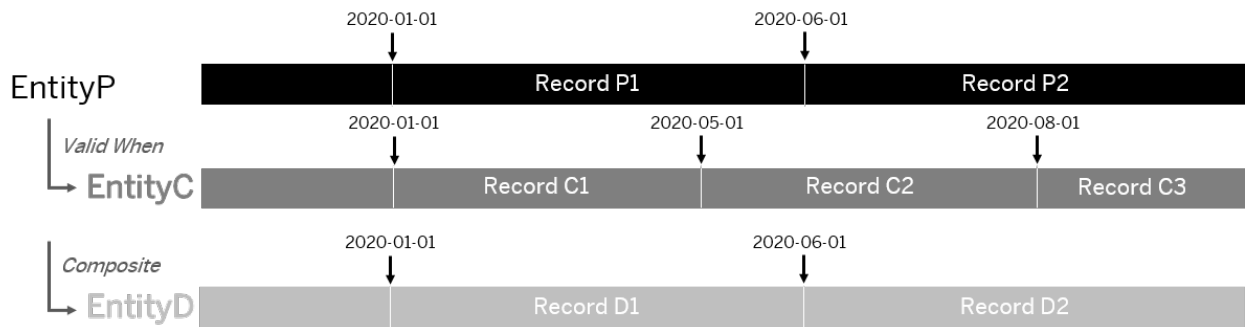
```
https://<API-Server>/odata/v2/TestObject('Key1')?$format=JSON&versionId=2
```

## 17.5.2.2 Sample Operations

### 17.5.2.2.1 Using `asOfDate` with `$expand`

Getting to know how `asOfDate` impacts the query results when it's used in combination with `$expand` to query effective-dated entities and its associations.

#### Example Entities



#### Composite Association

When querying an effective-dated entity and expanding to a composite association with `asOfDate`, the restrictive criteria of the parameter only works on the base object. However, as a composite association, EntityD has an `effectiveDate` type "from parent", the expanded data is also effective on this date.

#### Request

|             |   |
|-------------|---|
| Operation   | Query   |
| HTTP Method | GET   |
| URI         | <pre>https://&lt;API-server&gt;/odata/v2/EntityP? \$format=JSON&amp;\$expand=CompositeAssociation&amp;asOfDate=2020-08-15</pre> |

## Response

### Sample Code

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "https://<API-server>/odata/v2/EntityP(effectiveStartDate=datetime'2020-06-01T00:00:00',externalCode='P')",
          "type": "SFOData.cust_EntityP"
        },
        "externalCode": "P",
        "Record": "P2",
        "cust_ParentChildComposite": {
          "results": [
            {
              "__metadata": {
                "uri": "https://<API-server>/odata/v2/cust_EntityD(cust_EntityP_effectiveStartDate=datetime'2020-06-01T00:00:00',cust_EntityP_externalCode='P',externalCode='D')",
                "type": "SFOData.cust_EntityD"
              },
              "externalCode": "D",
              "Record": "D2"
            }
          ]
        }
      }
    ]
  }
}
```

## Valid-When Association

When querying an effective-dated entity and expanding to a valid-when association with the `&asOfDate`, the restrictive criteria of the parameter works on both the base object and the expanded data.

### Request

| Operation   | Query  |
|-------------|--|
| HTTP Method | GET  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/EntityP?\$format=JSON&amp;\$expand=ValidWhenAssociation&amp;asOfDate=2020-08-15</code> |

## Response

### Sample Code

```
{
  "d": {
    "results": [
      {

```

```

    "__metadata": {
      "uri": "https://<API-server>/odata/v2/
cust_EntityP(effectiveStartDate=datetime'2020-06-01T00:00:00',externalCode='P')",
      "type": "SFOData.cust_EntityP"
    },
    "externalCode": "P",
    "Record": "P2",
    "cust_ParentChildValidWhen": {
      "results": [
        {
          "__metadata": {
            "uri": "https://<API-server>/odata/v2/
cust_EntityC(effectiveStartDate=datetime'2020-08-01T00:00:00',externalCode='C')",
            "type": "SFOData.cust_EntityC"
          },
          "externalCode": "C",
          "Record": "C3"
        }
      ]
    }
  ]
}

```

## Navigation to a Composite Association and Referred Generic Object

In a more complex navigation scenario with `$expand` where you need to navigate to a composite association and its referred generic object, the date parameters filter both the base object and the referred generic object. In the following example, parameter `&fromDate=2018-10-01` works on both `PickListV2` and `values/parentPickListValueNav`:

```

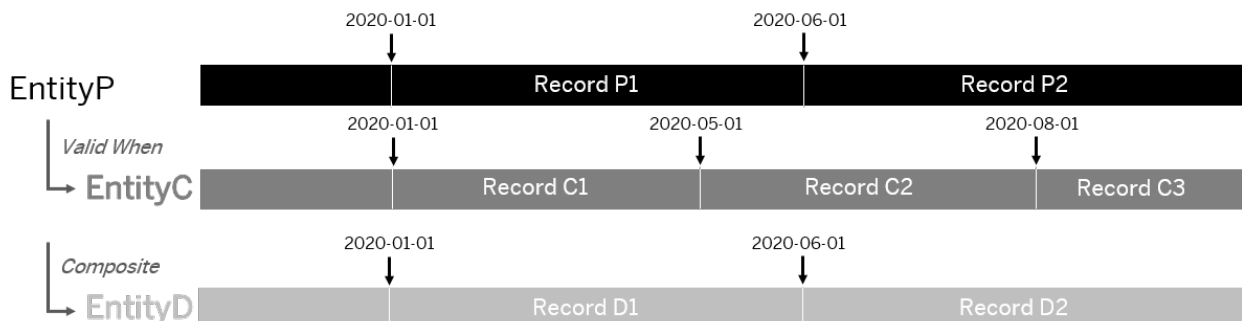
https://<hostname>/odata/v2/PickListV2?$format=JSON&$select=values/
parentPickListValueNav&$expand=values/parentPicklistValueNav&fromDate=2018-10-01

```

### 17.5.2.2 Using fromDate with \$expand

Getting to know how `&fromDate` impacts the query results when it's used in combination with `$expand` to query effective-dated entities and its associations.

#### Example Entities



## Composite Association

When querying an effective-dated entity and expanding to a composite association with `&fromDate`, the restrictive criteria of the parameter only works on the base object. However, as a composite association, EntityD has an effectiveDate type "from parent", the expanded data is also effective on this date.

### Request

|             |  |
|-------------|--|
| Operation   | Query  |
| HTTP Method | GET  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/EntityP?\$format=JSON&amp;\$expand=CompositeAssociation&amp;fromDate=2020-05-15</code> |

### Response

#### Sample Code

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "https://<API-server>/odata/v2/EntityP(effectiveStartDate=datetime'2020-01-01T00:00:00',externalCode='P')",
          "type": "SFOData.cust_EntityP"
        },
        "externalCode": "P",
        "Record": "P1",
        "cust_ParentChildComposite": {
          "results": [
            {
              "__metadata": {
                "uri": "https://<API-server>/odata/v2/EntityD(cust_EntityP_effectiveStartDate=datetime'2020-01-01T00:00:00',cust_EntityP_externalCode='P',externalCode='D')",
                "type": "SFOData.cust_EntityD"
              },
              "externalCode": "D",
              "Record": "D1"
            }
          ]
        }
      },
      {
        "__metadata": {
          "uri": "https://<API-server>/odata/v2/EntityP(effectiveStartDate=datetime'2020-06-01T00:00:00',externalCode='P')",
          "type": "SFOData.cust_EntityP"
        },
        "externalCode": "P",
        "Record": "P2",
        "cust_ParentChildComposite": {
          "results": [
            {
              "__metadata": {
                "uri": "https://<API-server>/odata/v2/EntityD(cust_EntityP_effectiveStartDate=datetime'2020-06-01T00:00:00',cust_EntityP_externalCode='P',externalCode='D')",
                "type": "SFOData.cust_EntityD"
              }
            }
          ]
        }
      }
    ]
  }
}
```





## Request

|             |   |
|-------------|---|
| Operation   | Query   |
| HTTP Method | GET   |
| URI         | https://<API-server>/odata/v2/EntityP?<br>\$format=JSON&\$expand=CompositeAssociation&toDate=2020-07-01 |

## Response

### Sample Code

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "https://<API-server>/odata/v2/EntityP(effectiveStartDate=datetime'2020-01-01T00:00:00',externalCode='P')",
          "type": "SFOData.cust_EntityP"
        },
        "externalCode": "P",
        "Record": "P1",
        "cust_ParentChildComposite": {
          "results": [
            {
              "__metadata": {
                "uri": "https://<API-server>/odata/v2/EntityD(EntityP_effectiveStartDate=datetime'2020-01-01T00:00:00',EntityP_externalCode='P',externalCode='D')",
                "type": "SFOData.cust_EntityD"
              },
              "externalCode": "D",
              "Record": "D1"
            }
          ]
        }
      },
      {
        "__metadata": {
          "uri": "https://<API-server>/odata/v2/EntityP(effectiveStartDate=datetime'2020-06-01T00:00:00',externalCode='P')",
          "type": "SFOData.cust_EntityP"
        },
        "externalCode": "P",
        "Record": "P2",
        "cust_ParentChildComposite": {
          "results": [
            {
              "__metadata": {
                "uri": "https://<API-server>/odata/v2/EntityD(EntityP_effectiveStartDate=datetime'2020-06-01T00:00:00',EntityP_externalCode='P',externalCode='D')",
                "type": "SFOData.cust_EntityD"
              },
              "externalCode": "D",
              "Record": "D2"
            }
          ]
        }
      }
    ]
  }
}
```

```
}
```

## Valid-When Association

When querying an effective-dated entity and expanding to a valid-when association with the `&toDate`, the restrictive criteria of the parameter works on both the base object and the expanded data.

### Request

| Operation   | Query  |
|-------------|--|
| HTTP Method | GET  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/EntityP?\$format=JSON&amp;\$expand=ValidWhenAssociation&amp;toDate=2020-07-01</code> |

### Response

#### Sample Code

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "https://<API-server>/odata/v2/EntityP(effectiveStartDate=datetime'2020-01-01T00:00:00',externalCode='P')",
          "type": "SFOData.cust_EntityP"
        },
        "externalCode": "P",
        "Record": "P1",
        "cust_ParentChildValidWhen": {
          "results": [
            {
              "__metadata": {
                "uri": "https://<API-server>/odata/v2/EntityC(effectiveStartDate=datetime'2020-01-01T00:00:00',externalCode='C')",
                "type": "SFOData.cust_EntityC"
              },
              "externalCode": "C",
              "Record": "C1"
            }
          ]
        }
      },
      {
        "__metadata": {
          "uri": "https://<API-server>/odata/v2/EntityP(effectiveStartDate=datetime'2020-06-01T00:00:00',externalCode='P')",
          "type": "SFOData.cust_EntityP"
        },
        "externalCode": "P",
        "Record": "P2",
        "cust_ParentChildValidWhen": {
          "results": [
            {
              "__metadata": {
                "uri": "https://<API-server>/odata/v2/EntityC(effectiveStartDate=datetime'2020-05-01T00:00:00',externalCode='C')",
```





The next example query contains only a date parameter. Without the `filterParentDate` parameter, the date parameter fails to work and is ignored. The result returns all available entries of the child entity:

```
https://<hostname>/odata/v2/EntityC?$format=JSON&toDate=2017-10-01
```

The last example query contains only the `filterParentDate` parameter. This will be treated as if the parent entity is filtered with the default date parameter, which equals to `&filterParentDate=true&asOfDate=<today's date>`. Therefore, the current effective entry Record C3 is returned.

```
https://<hostname>/odata/v2/EntityC?$format=JSON&filterParentDate=true
```

## 17.5.2.2.5 Querying Links of Effective Dated Entities With Composite Associations

You can use the `$links` option to query links of composite associations of an effective dated entity.

We use `PickListV2` and `PickListValueV2` as example.

### Request

| Operation   | Query   |
|-------------|---|
| HTTP Method | GET   |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/<br/>PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00',<br/>id='ExamplePicklist')/\$links/values</code> |

### Response

#### Sample Code

```
{
  "d": {
    "results": [
      {
        "uri": "https://<API-server>/odata/v2/  
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL  
istV2_id='ExamplePicklist',externalCode='ExampleValue1')"      },
      {
        "uri": "https://<API-server>/odata/v2/  
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL  
istV2_id='ExamplePicklist',externalCode='ExampleValue2')"      },
      {
        "uri": "https://<API-server>/odata/v2/  
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL  
istV2_id='ExamplePicklist',externalCode='ExampleValue3')"      },
      {
        "uri": "https://<API-server>/odata/v2/  
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL  
istV2_id='ExamplePicklist',externalCode='ExampleValue4')"      }
    ]
  }
}
```

```
}  
}
```

## 17.5.2.2.6 Filtering 1:M Composite Association

This topic shows examples of filtering parent and child entities whose association type is *Composite* and multiplicity is *One To Many*.

You can filter records of a parent entity based on properties of its child entities. When the parent and child entities have a one-to-many (1:M) relationship with its children, the query returns all parent records with full list of child entity records even though only one record of child entity meets the condition in the filter.

We use PickListV2 and PickListValueV2 as example entities. You can search for a picklist that contains a certain picklist value or you can search for a picklist whose parent picklist contains a certain picklist value.

### Example 1: Search for a Picklist That Contains a Certain Picklist Value

#### Request

| Operation   | Query  |
|-------------|--|
| HTTP Method | GET  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/PickList?<br/>\$format=JSON&amp;\$filter=values/externalCode eq<br/>'ABRAND' &amp;\$expand=values</code> |

#### Response

The query returns the picklist that has the said picklist value and a full list of child entity records if you choose to expand the association property.

#### Sample Code

```
{  
  "d": {  
    "results": [  
      {  
        "__metadata": {  
          "uri": "https://<api-server>/odata/v2/  
PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00',id='CARBRAND')",  
          "type": "SFOData.PickListV2"  
        },  
        "id": "CARBRAND",  
        "status": "A",  
        "values": {  
          "results": [  
            {  
              "__metadata": {  
                "uri": "https://<api-server>/odata/v2/  
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL  
istV2_id='CARBRAND',externalCode='ABRAND')",
```



## Example 2: Search for a Picklist Whose Parent Picklist Contains a Certain Picklist Value

### Request

| Operation   | Query   |
|-------------|---|
| HTTP Method | GET   |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/PickList?<br/>\$format=JSON&amp;\$filter=parentPickListNav/values/externalCode<br/>eq<br/>'ABRAND'&amp;\$expand=values,parentPickListNav,parentPickListNav<br/>/values</code> |

### Response

The query returns the picklist whose parent picklist has the said picklist value and a full list of child entity records if you choose to expand association properties.

#### Sample Code

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "https://<api-server>/odata/v2/  
PickListV2(effectiveStartDate=datetime'2020-07-01T00:00:00',id='CARCOLOR')",
          "type": "SFOData.PickListV2"
        },
        "id": "CARCOLOR",
        "parentPickList": "CARBRAND",
        "status": "A",
        "values": {
          "results": [
            {
              "__metadata": {
                "uri": "https://<api-server>/odata/v2/  
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-01T00:00:00',PickL  
istV2_id='CARCOLOR',externalCode='Black')",
                "type": "SFOData.PickListValueV2"
              },
              "externalCode": "Black",
              "status": "A"
            },
            {
              "__metadata": {
                "uri": "https://<api-server>/odata/v2/  
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-01T00:00:00',PickL  
istV2_id='CARCOLOR',externalCode='Yellow')",
                "type": "SFOData.PickListValueV2"
              },
              "externalCode": "Yellow",
              "status": "A"
            }
          ]
        }
      },
      {
        "__metadata": {
          "uri": "https://<api-server>/odata/v2/  
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-01T00:00:00',PickL  
istV2_id='CARCOLOR',externalCode='Orange')",
          "type": "SFOData.PickListValueV2"
        }
      }
    ]
  }
}
```

```

        "type": "SFOData.PickListValueV2"
    },
    "externalCode": "Orange",
    "status": "A"
}
]
},
"parentPickListNav": {
    "__metadata": {
        "uri": "https://<api-server>/odata/v2/
PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00',id='CARBRAND')",
        "type": "SFOData.PickListV2"
    },
    "id": "CARBRAND",
    "values": {
        "results": [
            {
                "__metadata": {
                    "uri": "https://<api-server>/odata/v2/
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL
istV2_id='CARBRAND',externalCode='ABRAND')",
                    "type": "SFOData.PickListValueV2"
                },
                "externalCode": "ABRAND"
            },
            {
                "__metadata": {
                    "uri": "https://<api-server>/odata/v2/
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL
istV2_id='CARBRAND',externalCode='B BRAND')",
                    "type": "SFOData.PickListValueV2"
                },
                "externalCode": "B BRAND"
            },
            {
                "__metadata": {
                    "uri": "https://<api-server>/odata/v2/
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL
istV2_id='CARBRAND',externalCode='C BRAND')",
                    "type": "SFOData.PickListValueV2"
                },
                "externalCode": "C BRAND"
            },
            {
                "__metadata": {
                    "uri": "https://<api-server>/odata/v2/
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL
istV2_id='CARBRAND',externalCode='D BRAND')",
                    "type": "SFOData.PickListValueV2"
                },
                "externalCode": "D BRAND"
            },
            {
                "__metadata": {
                    "uri": "https://<api-server>/odata/v2/
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL
istV2_id='CARBRAND',externalCode='E BRAND')",
                    "type": "SFOData.PickListValueV2"
                },
                "externalCode": "E BRAND"
            },
            {
                "__metadata": {
                    "uri": "https://<api-server>/odata/v2/
PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickL
istV2_id='CARBRAND',externalCode='F BRAND')",
                    "type": "SFOData.PickListValueV2"
                },
                "externalCode": "F BRAND"
            }
        ]
    }
}

```

```
    "externalCode": "FBRAND"
  }
]
}
}
}
}
}
```

## Related Information

[\\$filter](#)

### 17.5.2.2.7 Retrieving Workflow Information for Pending Data

You can use the `wfRequestNav` navigation property and the `recordStatus` parameter to retrieve workflow information for pending data of MDF entities.

When an MDF Generic Object (GO) with workflow routing configured is exposed to OData, a navigation property named `wfRequestNav` is automatically added to the OData API entity. This navigation property allows you to retrieve the corresponding workflow information for pending data. Although `wfRequestNav` is available by default for all MDF OData API entities with a workflow routing, you can only expand it if the [Pending Data](#) option is enabled in the object definition.

#### Note

To enable workflow and pending data, associate the MDF generic object with a workflow and set the [Pending Data](#) field to **Yes** in [Configure Object Definitions](#).

Child entities in a composite association don't have the `wfRequestNav` property.

Since MDF OData API only supports workflow information on pending data, you must add the `recordStatus` parameter to specifically filter the pending records, and then expand the workflow information. The following examples show how to do this.

You can use `&recordStatus=pending` to filter all pending records of the entity, and then expand the workflow information:

```
https://<API-endpoint-URL>/odata/v2/cust_MDFEntity?
$format=JSON&recordStatus=pending&$expand=wfRequestNav
```

If you use a key predicate to query a single record, even if it's a pending record, you still need `&recordStatus=pending` to be able to expand the workflow information:

```
https://<API-endpoint-URL>/odata/v2/cust_MDFEntity('CC01')?
$format=JSON&recordStatus=pending&$expand=wfRequestNav
```

## Error Codes

| Error Code              | Description  |
|-------------------------|--|
| COE_UNSUPPORTED_FEATURE | <p>[COE0025]Unsupported feature: Expand workflow navigation from normal data is NOT supported.</p> <p>If you try to expand wfRequestNav on a normal entry, this error occurs. wfRequestNav can only be expanded if it's a pending entry.</p> |

## Related Information

[recordStatus \[page 235\]](#)

### 17.5.2.2.8 Querying Effective-Dated Entities Using lastModifiedDateTime

You can use the lastModifiedDateTime field in a \$filter option to query the records of an effective-dated entity.

The value of lastModifiedDateTime in a request URI is in date/time format without timezone information:

```
datetime'yyyy-mm-ddThh:mm:ss'
```

For example: `$filter=lastModifiedDateTime gt datetime'2019-09-01T12:00:00'`.

You can use lastModifiedDateTime together with other time parameters such as fromDate, toDate, and asOfDate. You can also reverse the \$filter condition by placing the date/time value before lastModifiedDateTime to get different query results.

## Using lastModifiedDateTime with Time Parameters

When you use both lastModifiedDateTime and a date parameter in the same request, the following rules apply:

- The lastModifiedDateTime filter finds all **external keys** that have at least one record satisfies the filter condition.
- The time parameter then filters all **records** under the external keys and returns the ones that satisfy the condition in the result.

This means that in the returned result, the lastModifiedDateTime value may or may not satisfy the filter condition in the request. The following example explains how it works:

#### Example

```
https://<API-Server>/odata/v2/EmpJob?$format=JSON&$filter=lastModifiedDateTime  
gt datetime'2017-01-01T00:00:00'&toDate=2019-01-01
```



In this example, you try to find all employees who have their job information changed after 2017-01-01 and pull their entire job history, including the ones that were last modified before 2017-01-01.

## Placing lastModifiedDateTime Before or After DateTime Value

The position of date/time value in the lastModifiedDateTime filter determines how the server interprets the query. When the date/time value is placed after lastModifiedDateTime, the filter finds all external keys that satisfy the condition. When the date/time value is placed before lastModifiedDateTime, the filter works as you normally understand it: it returns all **records** with last modified date/time satisfying the filter.

In the previous example, if you want to display only the job history that were last modified after 2017-01-01, you can change the query by moving the date/time value before lastModifiedDateTime in the filter:

```
https://<API-Server>/odata/v2/EmpJob?  
$format=JSON&$filter=datettime'2017-01-01T00:00:00' gt  
lastModifiedDateTime&toDate=2019-01-01
```

## Related Information

[Using the DateTime Format](#)

[asOfDate \[page 232\]](#)

[fromDate \[page 233\]](#)

[toDate \[page 234\]](#)

## 17.5.3 Edit Operations

Parameters and examples of edit operations for MDF OData API entities, including insert, upsert, replace, merge, and delete operations.

### MDF Upsert Operation

As of Q4 2019 Release, upsert operations on all MDF OData entities and MDF Foundation Objects can return business keys in the response if the keys are provided in the request. That includes both parent objects and child objects. This information helps you identify which record has been successfully upserted and which has failed.

## 17.5.3.1 Edit Parameters

### 17.5.3.1.1 workflowConfirmed

Use parameter `workflowConfirmed` to control whether or not the insert or upsert operation triggers a workflow.

#### Prerequisites

Either you've defined the workflow routing for the MDF object or you've associated the object with a business rule that triggers a workflow.

The API user doesn't have the [Admin access to MDF OData API](#) permission.

#### ⚠ Caution

The parameter has no impact on admin users with the [Admin access to MDF OData API](#) permission. If an admin user creates or changes a record for a workflow-enabled entity, no workflow is triggered and the record is saved as normal data.

The parameter doesn't work with MDF Foundation Objects.

#### What Happens If Two Workflows Are Defined?

If the MDF object is associated with one workflow in the [Workflow Routing](#) field and another workflow through a business rule, the business rule-based workflow supersedes when the rule is triggered.

#### 🔗 Example

Object `cust_ABC` is associated with two workflows: workflow **Manager** from object definition and workflow **EmployeeHR** from a business rule. When you change data that doesn't trigger the business rule, the **Manager** workflow applies. When you change data that triggers the business rule, the **EmployeeHR** workflow applies.

#### Parameter Values

| Value                        | Description                 |
|------------------------------|-----------------------------|
| <code>false</code> (default) | A workflow isn't triggered. |
| <code>true</code>            | A workflow is triggered.    |

## Use Case 1: Upsert a Record With `workflowConfirmed=true`

A typical scenario to use the `workflowConfirmed` parameter is on the UI where you create or edit a record. When you save the record, a dialog pops up for you to submit the changes for approval. If you submit, an API request is sent with `workflowConfirmed=true`:

### Request

|             |  |
|-------------|--|
| Operation   | Upsert   |
| HTTP Method | POST   |
| URI         | <code>https://&lt;API-Server&gt;/odata/v2/upsert?workflowConfirmed=true</code>   |
| Payload     | <pre>{   "__metadata": {     "uri": "cust_TestObject('TO-001')"   },   "externalCode": "TO-001",   "externalName": "Test Object 001" }</pre> |

### Response

If the upsert operation is successful, a workflow request is created.

```
{
  "d": [
    {
      "key": null,
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}
```

## Use Case 2: Upsert a Record With `workflowConfirmed=false`

As a user without the [Admin access to MDF OData API](#) permission, if you upsert a workflow-enabled entity without the `workflowConfirmed` parameter or set the parameter value to `false`, you get an error:

### Request

|             |        |
|-------------|--------|
| Operation   | Upsert |
| HTTP Method | POST   |

URI `https://<API-Server>/odata/v2/upsert?workflowConfirmed=false`

Payload

```
{
  "__metadata":{
    "uri":"cust_TestObject('TO-002')",
    "externalCode":"TO-002",
    "externalName":"Test Object 002"
  }
}
```

## Response

```
{
  "d": [
    {
      "key": null,
      "status": "ERROR",
      "editStatus": null,
      "message": "To trigger workflow, please set the 'workflowConfirmed'
flag to true. with the index 0",
      "index": 0,
      "httpCode": 500,
      "inlineResults": null
    }
  ]
}
```

### 17.5.3.1.2 warningAcknowledged

Use parameter `warningAcknowledged` to skip warning messages during upsert of an MDF record.

## Prerequisites

A business rule has been defined for the MDF object to raise a warning message.

The API user doesn't have the [Admin access to MDF OData API](#) permission.

### Note

The parameter has no impact on admin users with the [Admin access to MDF OData API](#) permission. If an admin user creates or changes a record for an entity that triggers a warning message, the warning is skipped.

## Parameter Values

| Value                        | Description  |
|------------------------------|--|
| <code>false</code> (default) | If the business rule is triggered, a warning message appears and the upsert fails.       |
| <code>true</code>            | No warning message appears even if the business rule is triggered. Upsert is successful. |

### Use Case 1: Upsert a Record With `warningAcknowledged=true`

A typical scenario to use the `warningAcknowledged` parameter is on the UI where you create or edit a record. When you save the record, a warning message pops up. If you choose OK, the warning is acknowledged and an API request is sent with `warningAcknowledged=true`.

#### Request

|             |  |
|-------------|--|
| Operation   | Upsert   |
| HTTP Method | POST   |
| URI         | <code>https://&lt;API-endpoint-URL&gt;/odata/v2/upsert&amp;warningAcknowledged=true</code>   |
| Payload     | <pre>{   "__metadata": {     "uri": "cust_TestObject('TO-001')",     "externalCode": "TO-001",     "externalName": "Test Object 001"   } }</pre> |

#### Response

If the upsert is successful, a record is created.

```
{
  "d": [
    {
      "key": null,
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}
```

## Use Case 2: Upsert a Record With `warningAcknowledged=false`

As a user without the [Admin access to MDF OData API](#) permission, if you upsert a warning-enabled entity without the `warningAcknowledged` parameter or set the parameter value to `false`, you get an error:

### Request

|             |   |
|-------------|---|
| Operation   | Upsert  |
| HTTP Method | POST  |
| URI         | <code>https://&lt;API-endpoint-URL&gt;/odata/v2/upsert&amp;warningAcknowledged=false</code>   |
| Payload     | <pre>{   "__metadata":{     "uri":"cust_TestObject('TO-002')",   },   "externalCode":"TO-002",   "externalName":"Test Object 002" }</pre> |

### Response

```
{
  "d": [
    {
      "key": null,
      "status": "ERROR",
      "editStatus": null,
      "message": "<user-defined message text>; To be able to save MDF object with warning, please set the 'warningAcknowledged' flag to true. with the index 0",
      "index": 0,
      "httpCode": 500,
      "inlineResults": null
    }
  ]
}
```

## 17.5.3.2 Sample Operations

### 17.5.3.2.1 Inserting An Entry to A Composite Child Entity

We use PickListValueV2 as the example of a composite child entity and show you how to insert an entry.

#### Use Case: Create a Picklist Value

##### Request

|             |   |
|-------------|---|
| Operation   | Insert  |
| HTTP Method | POST  |
| URI         | https://<API-server>/odata/v2/PickListValueV2 |

##### Payload

```
{
  "__metadata": {
    "uri":
    "PickListValueV2(PickListV2_effectiveStartDate=datetime'1900-01-01T00:00:00',PickListV2_id='OfficeLocation',externalCode='717011')",
    "type": "SFOData.PickListValueV2"
  },
  "externalCode": "717011",
  "PickListV2_effectiveStartDate": "/Date(-2208988800000)/",
  "PickListV2_id": "OfficeLocation",
  "status": "A",
  "label_en_US": "Guangzhou"
}
```

##### Note

You must include at least one label property in the request payload.

##### Response

```
{
  "d": {
    "__metadata": {
      "uri": "https://<API-server>/odata/v2/PickListValueV2(PickListV2_effectiveStartDate=datetime'1900-01-01T00:00:00',PickListV2_id='OfficeLocation',externalCode='717011')",
      "type": "SFOData.PickListValueV2"
    },
    "status": "A",
    "label_en_US": "Guangzhou",
    "PickListV2_id": "OfficeLocation",
    "PickListV2_effectiveStartDate": "/Date(-2208988800000)/",
    "externalCode": "717011"
  }
}
```

## 17.5.3.2.2 Updating Picklist Values Using Upsert with Purge Type Parameters

You can use the `upsert` operation with the `purgeType` parameters to perform a full purge or an incremental update on an entity.

We use `PickListValueV2` as the example of a composite child entity and show you how to fully purge picklist values or update a picklist value of a picklist.

### Deleting All Picklist Values

#### Request

|             |   |
|-------------|---|
| Operation   | Upsert  |
| HTTP Method | POST  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/upsert?purgeType=full</code>  |
| Payload     | <pre>{   "__metadata": {     "uri": "PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00',id='ExistingPickList')"   },   "status": "A" }</pre> |

#### Response

##### Sample Code

```
{
  "d": [
    {
      "key": "PickListV2/
effectiveStartDate=2020-07-17T00:00:00.000-04:00,PickListV2/id=ExistingPickList",
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}
```

The picklist now has no picklist values.



## Replacing Old Picklist Values with New Ones

### Request

|             |  |
|-------------|--|
| Operation   | Upsert   |
| HTTP Method | POST   |
| URI         | https://<API-server>/odata/v2/upsert?purgeType=full  |
| Payload     | <pre>{   "__metadata": {     "uri": "PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00',id='PickListExample')"   },   "status": "A",   "values": [     {       "__metadata": {         "uri": "PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickListV2_id='PickListExample',externalCode='NEWVALUE1')"       },       "status": "A",       "label_en_US": "New Value 1"     },     {       "__metadata": {         "uri": "PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickListV2_id='PickListExample',externalCode='NEWVALUE2')"       },       "status": "A",       "label_en_US": "New Value 2"     }   ] }</pre> |

### Response

#### Sample Code

```
{
  "d": [
    {
      "key": "PickListV2/
effectiveStartDate=2020-07-17T00:00:00.000-04:00,PickListV2/id=PickListExample",
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}
```

All previous picklist values of the picklist are replaced with the two new values.

## Updating the Label of an Existing Picklist Value

### Request

|             |   |
|-------------|---|
| Operation   | Upsert  |
| HTTP Method | POST  |
| URI         | https://<API-server>/odata/v2/upsert?purgeType=incremental<br>or https://<API-server>/odata/v2/upsert   |
| Payload     | <pre>{   "__metadata": {     "uri": "PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00',id='PickListExample')"   },   "status": "A",   "values": [ {     "__metadata": {       "uri": "PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickListV2_id='PickListExample',externalCode='NEWVALUE1')"     },     "status": "A",     "label_en_US": "New Label"   }   ] }</pre> |

### Response

#### Sample Code

```
{
  "d": [
    {
      "key": "PickListV2/effectiveStartDate=2020-07-17T00:00:00.000-04:00,PickListV2/id=PickListExample",
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}
```

Only the label of picklist value "NEWVALUE1" is updated.

## Related Information

[Upsert](#)

[Edit Parameter: purgeType](#)

### 17.5.3.2.3 Updating Picklist Values Using Merge

You can use the merge operation to incrementally update a property without providing all required properties in the payload.

We use `PickListValueV2` as the example of a composite child entity and show you how to update picklist values using merge.

#### Request

|             |   |
|-------------|---|
| Operation   | Merge   |
| HTTP Method | POST  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickListV2_id='ExamplePicklist',externalCode='OldValue')</code> |
| Headers     | <code>x-http-method: MERGE</code>   |
| Payload     | <pre>{   "externalCode": "NewValue" }</pre>   |

#### Response

A successful operation returns status 200 OK with no payload.

#### Related Information

[Merge](#)

### 17.5.3.2.4 Replacing Picklist Using Replace

You can use the replace operation to update an existing record with the fields specified in the request payload and resets all other fields to their default value. Unlike the merge operation, you must provide all required fields in the request payload.

We use `PickListV2` as an example to show you how to replace an old picklist with a new one. The example picklist had an old name and its display order was numeric.

#### Request

|           |         |
|-----------|---------|
| Operation | Replace |
|-----------|---------|

|             |  |
|-------------|--|
| HTTP Method | PUT  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/<br/>PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00'<br/>,id='OldID')</code> |
| Payload     | <pre>{   "id": "NewID",   "status": "A" }</pre>  |

## Response

A successful operation returns status 200 OK with no payload. In the updated version, the example picklist has no name and its display order is reset to the default value that is alphabetic.

## Related Information

[Replace](#)

### 17.5.3.2.5 Deleting Picklist or Picklist Value

You can use the delete operation to delete an entry by passing the key predicate in the request URI.

We use `PickListV2` and `PickListValueV2` as examples to show you how to delete an entry.

#### Request

Deleting a Picklist

|             |  |
|-------------|--|
| Operation   | Delete   |
| HTTP Method | DELETE   |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/<br/>PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00'<br/>,id='ExamplePicklist')</code> |

Deleting a Picklist Value

|             |  |
|-------------|--|
| Operation   | Delete   |
| HTTP Method | DELETE   |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/<br/>PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickListV2_id='ExamplePicklist',externalCode='ExampleValue')</code> |

## Response

A successful delete operation returns status 200 OK with no response payload. The picklist and picklist value specified in the URI are deleted from the system.

## 17.5.3.2.6 Creating Links for MDF Entities

You can use the `$links` option to create links for MDF entities.

### Rules for Creating Links for MDF Entities

The following rules apply when you create links between MDF entities (including MDF Foundation Objects):

- You can't create links for MDF Generic Objects with composite type associations.
- A new link is always inserted for a 1:n association. If the same link already exists, the POST operation returns `204 No Content` without data change.
- For 1:1 relationship, if a link already exists, the POST operation fails.

### Adding a Parent Picklist Using Links

We use `PickListV2` as the example to show you how to create links for MDF entities. You can't create a new picklist value by using the `$links` option because the association between `PickListV2` and `PickListValueV2` is composite.

#### Request

|             |   |
|-------------|---|
| HTTP Method | POST  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/<br/>PickListV2(effectiveStartDate=datetime'2020-07-17T00:00:00'<br/>,id='ChildPicklist')/\$links/parentPickListNav</code>                                  |
| Payload     | <div style="border: 1px solid #ccc; padding: 10px;"><p>↔ Sample Code</p><pre>{<br/><br/>  "uri": "PickListV2(effectiveStartDate=datetime'2020-07-10T00:00:00',id='ParentPickList')"<br/>}</pre></div> |

#### Response

A successful operation returns status 204 (No Content) with no payload. In the updated version, the child picklist is assigned with a parent picklist.

## Related Information

[Working with Links](#)

### 17.5.3.2.7 Updating Parent Picklist Value Using Links

You can use the `$links` option to update links of noncomposite associations.

We use `PickListValueV2` as the example and updates the parent picklist value.

#### Note

You can't update picklist values of a picklist using the `$links` option because the association type between `PickListV2` and `PickListValueV2` is composite.

#### Request

|             |  |
|-------------|--|
| HTTP Method | PUT  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/<br/>PickListValueV2(PickListV2_effectiveStartDate=datetime'2020-07-17T00:00:00',PickListV2_id='ChildPicklist',externalCode='ExampleValue')/\$links/<br/>parentPickListValueNav(PickListV2_effectiveStartDate=datetime'2020-07-10T00:00:00',PickListV2_id='ParentPicklist',externalCode='OldValue')</code> |
| Payload     | <div><p>↔ Sample Code</p><pre>{<br/><br/>  "uri": "PickListV2(PickListV2_effectiveStartDate=datetime'2020-07-10T00:00:00',PickListV2_id='ParentPicklist',externalCode='NewValue')"<br/>}</pre></div>   |

#### Response

A successful operation returns status 204 (No Content) with no payload. In the updated version, the value of the child picklist is assigned with a new value from the parent picklist.

## Related Information

[Working with Links](#)

## 17.5.3.2.8 Linking Attachments to an MDF Entity

How to link an attachment to an MDF OData API entity.

### Context

The attachment navigation property in an MDF entity doesn't allow you to inline edit attachments. To link an attachment to an MDF entity, you first create the attachment object with the [Attachment](#) entity, and then link this object to the MDF entity.

#### ⓘ Note

You're not allowed to add the same attachment ID to multiple objects. If you want to link the same attachment to another object, upload it again to create a new attachment ID.

### Procedure

1. Create an attachment object with the [Attachment](#) entity.

Sample Request:

|             |   |
|-------------|---|
| Operation   | Upsert  |
| HTTP Method | POST  |
| URI         | <code>https://&lt;API-Server&gt;/odata/v2/upsert</code> |

Payload

```
{
  "__metadata": {
    "uri": "Attachment"
  },
  "fileName": "Resume.pdf",
  "module": "RECRUITING",
  "userId": "cgrant",
  "viewable": true,
  "fileContent":
  "PHA+VGhp...KSHlsHcq9JS"
}
```

Sample Response:

```
{
  "d": [
    {
      "key": "Attachment/attachmentId=499",
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": "Upserted successfully",
      "index": 0,
      "httpCode": 200,
    }
  ]
}
```

```
    "inlineResults": null
  }
]
}
```

2. Create a new entry or update an existing entry for the MDF entity with the new attachment created in the previous step.

Sample Request:

|             |  |
|-------------|--|
| Operation   | Upsert   |
| HTTP Method | POST   |
| URI         | https://<API-Server>/odata/v2/upsert   |
| Payload     | <pre>{   "__metadata": {     "uri": "cust_TestMDFObject"   },   "externalCode": "test entry 01",   "effectiveStartDate": "/Date(1565222400000)/",   "externalName": "Test MDF Entry 01",   "cust_attachmentNav": {     "__metadata": {       "uri": "Attachment(499L)"     }   } }</pre> |

### 17.5.3.2.9 Creating Self-Referenced Record Using Upsert

You can use the `upsert` operation to create a self-referenced record for custom entities whose effective dating type isn't MCPD.

#### Example Entity

We use `cust_EntityP` as the example entity.

- `cust_EntityP` is a custom entity.
- `cust_EntityP` has a *Generic Object* field `cust_goNav` and the *Valid Value Source* is `cust_EntityP` itself.
- The effective dating type of `cust_EntityP` isn't MCPD.



## Request

|             |   |
|-------------|---|
| Operation   | Upsert  |
| HTTP Method | POST  |
| URI         | https://<API-server>/odata/v2/upsert  |
| Payload     | <pre>{   "__metadata": {     "uri":     "cust_EntityP(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueA')"   },   "cust_goNav": {     "__metadata": {       "uri":       "cust_EntityP(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueA')"     },     "externalCode": "ValueA"   } }</pre> |

## Response

### Sample Code

```
{
  "d": [
    {
      "key": "cust_EntityP/
effectiveStartDate=2020-07-17T00:00:00Z,cust_EntityP/externalCode=ValueA",
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}
```

## Wrong Method to Create Self-Referenced Record Using Insert

You can't create self-referenced record using insert.

## Request

|             |   |
|-------------|---|
| Operation   | Insert  |
| HTTP Method | POST  |
| URI         | https://<API-server>/odata/v2/cust_EntityP  |
| Payload     | <pre>{   "__metadata": {     "uri":     "cust_EntityP(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueA')"   },   "cust_goNav": {     "__metadata": {       "uri":       "cust_EntityP(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueA')"     },     "externalCode": "ValueA"   } }</pre> |

## Response

### Sample Code

```
{
  "error": {
    "code": "COE_GENERAL_BAD_REQUEST",
    "message": {
      "lang": "en-US",
      "value": "[COE0018]Record '(effectiveStartDate=2020-07-17, externalCode=ValueA)' of type 'cust_EntityP' already exists."
    }
  }
}
```

## 17.5.3.2.10 Upsert Self-Referenced Entity with Different Field Value in Payload

Refer to the sample operation for how the system handles different values of the self-referenced field in payload.

## Example Entity

We use `cust_EntityP` as the example entity.

- `cust_EntityP` is a custom entity.
- `cust_EntityP` has a *Generic Object* field `cust_goNav` and the *Valid Value Source* is `cust_EntityP` itself.
- The effective dating type of `cust_EntityP` isn't MCPD.

## Request

|             |   |
|-------------|---|
| Operation   | Upsert  |
| HTTP Method | POST  |
| URI         | https://<API-server>/odata/v2/upsert  |
| Payload     | <pre>{   "__metadata": {     "uri":     "cust_EntityP(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueA')"   },   "name": "A",   "cust_goNav": {     "__metadata": {       "uri":       "cust_EntityP(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueA')"     },     "name": "B"   } }</pre> |

## Response

The *name* of cust\_EntityP's record **ValueA** is updated as **A**.

### Sample Code

```
{
  "d": [
    {
      "key": "cust_EntityP/
effectiveStartDate=2020-07-17T00:00:00Z,cust_EntityP/externalCode=ValueA",
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}
```

## 17.5.3.2.11 Upsert Handling Different Field Values in Payload

Refer to the sample operation for how the system handles different values of the same *Generic Object* field in payload.

### Example Entity

We use `cust_EntityP` as the example entity.

- `cust_EntityP` is a custom entity.
- `cust_EntityP` has a *Generic Object* field `cust_goNav` and the *Valid Value Source* is `cust_EntityC`.
- The effective dating type of both `cust_EntityP` and `cust_EntityC` isn't MCPD.

### Request

|             |   |
|-------------|---|
| Operation   | Upsert  |
| HTTP Method | POST  |
| URI         | <code>https://&lt;API-server&gt;/odata/v2/upsert</code> |

Payload

```
[
  {
    "__metadata": {
      "uri":
      "cust_EntityP(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueA')"
    },
    "cust_goNav": {
      "__metadata": {
        "uri":
        "cust_EntityC(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueB')"
      },
      "name": "Name A"
    }
  },
  {
    "__metadata": {
      "uri":
      "cust_EntityP(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueA')"
    },
    "cust_goNav": {
      "__metadata": {
        "uri":
        "cust_EntityC(effectiveStartDate=datetime'2020-07-17T00:00:00',externalCode='ValueB')"
      },
      "name": "Name B"
    }
  }
]
```

## Response

The *name* of cust\_EntityC's record **ValueB** is updated as **Name B**.

### Sample Code

```
{
  "d": [
    {
      "key": "cust_EntityP/
effectiveStartDate=2020-07-17T00:00:00Z,cust_EntityP/externalCode=ValueA",
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 0,
      "httpCode": 200,
      "inlineResults": null
    },
    {
      "key": "cust_EntityP/
effectiveStartDate=2020-07-17T00:00:00.000Z,cust_EntityP/externalCode=ValueA",
      "status": "OK",
      "editStatus": "UPSERTED",
      "message": null,
      "index": 1,
      "httpCode": 200,
      "inlineResults": null
    }
  ]
}
```

```
}  
  ]  
}
```

# 18 Migrating MDF Configurations Using the Instance Sync Tool

## 18.1 Instance Sync Tool

This section explains how to use the Instance Sync Tool to migrate their configurations from one instance to another.

Customers can use the instance sync tool to migrate their configurations from one instance to another. The sync tool acts as a push model. The source instance selectively chooses configuration objects to push to the target instance.

Once enabled in Provisioning, you have admin permissions to decide who has access to the sync tool. Permissions can be given at object level. Having access to one object gives you access to the sync tool.

### → Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

You can transfer the following configuration objects in the sync tool:

- MDF Generic Objects
- MDF Configurable UI
- MDF Object Definition
- MDF Rules

### 18.1.1 Transferring Configurations Between Instances

This sections explains how to transfer your configurations between system instances.

All configuration syncs are triggered from the source instance. If you have permission granted through the administrative privileges, you can use the tool in the source instance.

#### Select Target Instance

The admin must select the target instance to push the configuration set to single select. You can have access to multiple target instances. However, you can only push configurations one instance at a time.

#### Select Objects and Settings

Each object can be selected and you have granular control to decide how the transfer must happen. The tool supports the selection of multiple objects.

## 18.1.2 Transferring Generic Objects

This sections explains how to transfer MDF generic objects between system instances.

### Selection

The admin can select multiple MDF generic objects.

### Options

The admin can specify whether an existing MDF generic object in the target instance is overwritten or ignored.

### Copy Logic

The tool can copy all content within the selected MDF generic objects.

## 18.1.3 Transferring Configurable UIs

This sections explains how to transfer configurable UIs between system instances.

### Selection

The admin can select multiple MDF configurable UIs.

### Options

The admin can specify whether an existing MDF configurable UI in the target instance is overwritten or ignored.

### Copy Logic

The tool can copy all content within the selected MDF configurable UIs.

## 18.1.4 Transferring Object Definitions

This sections explains how to transfer MDF object definitions between system instances.

### Selection

The admin can select multiple MDF object definitions to be migrated across instances.

### Options

The admin can specify whether an existing MDF object definition in the target instance is overwritten or ignored.

### Copy Logic

The tool can copy all content within the selected MDF object definition. If the underlying rules are synced, the object definition is also synced so that the object definition and rules are never out of sync.



## 18.1.5 Transferring MDF Rules

This sections explains how to transfer MDF rules between system instances.

### Selection

The admin can select multiple rules.

### Options

The admin can specify whether an existing MDF rule in the target instance is overwritten or ignored.

### Copy Logic

The tool can copy all content within the selected rules.

## Example

### Scenario A

Source: Object Definition A with Rule 1 and Rule 2 exist in the source instance.

Target: Object Definition A with Rule 1 and Rule 2 do not exist in the target instance.

| Case  | System Behavior in Target  |
|---|--|
| Perform instance sync by selecting Object Definition A with Rule 1 and Rule 2.  | Object Definition A associated with Rule 1 and Rule 2 is synced.   |
| Modify Rule 1 and perform instance sync by selecting Rule 1.  | Rule 1 is synced. Object Definition A is also synced because the underlying Rule 1 is synced.                          |
| Add new Rule 3 and associate it with Object Definition A in source instance, and then perform instance sync by selecting Rule 3 only. | Rule 3 is synced. Object Definition A is also synced and associated with Rule 3 because its underlying rule is synced. |

### Scenario B

Source: Object Definition A with Rule 1 and Rule 2 exist in the source instance.

Target: Object Definition A with Rule 1 exist in the target instance. Rule 2 does not exist in the target instance.

| Case   | System Behavior in Target  |
|--|--|
| Perform instance sync by selecting Object Definition A and Rule 2. | Object Definition A associated with Rule 1 and Rule 2 is synced.   |
| Perform instance sync by selecting Object Definition A.            | Object Definition A sync fails because all associated rules cannot be synced. The information is included in the Instance Sync Monitor report. |

Perform instance sync by selecting Rule 2.

Rule 2 is synced. Object Definition A is also synced and associated with Rule 2 because its underlying rule is synced.

## 18.2 Test Sync/Run Sync

This section explains how to simulate and actually run the sync tool.

### Test Sync

You can simulate the sync without actually transferring any configurations. To simulate the sync, choose [Test Sync](#). You can check the simulation results in the instance synchronization monitor tool. For information about the monitor tool, see [Monitoring a Sync Operation \[page 280\]](#).

| Artifact           | Sub Type | Number of Selections | Duplicate Resolution |
|--------------------|----------|----------------------|----------------------|
| Dashboard Settings | -        | 3                    | OVERWRITE            |

### Run Sync

To run the sync, choose [Run Sync Now](#). You can check the results in the instance synchronization monitor tool. For information about the monitor tool, see [Monitoring a Sync Operation \[page 280\]](#).

#### Note

Multiple administrators cannot run a sync operation simultaneously. There can only be one sync selection wizard in progress.

You will receive a warning message during the sync if the instances are running on different release versions, for example, if one data center is received the quarterly update but the other hasn't.

| Artifact           | Sub Type | Number of Selections | Duplicate Resolution |
|--------------------|----------|----------------------|----------------------|
| Dashboard Settings | -        | 3                    | OVERWRITE            |

Exit and Finish Later      Back      **Run Sync Now** ▶      Test Sync ▶

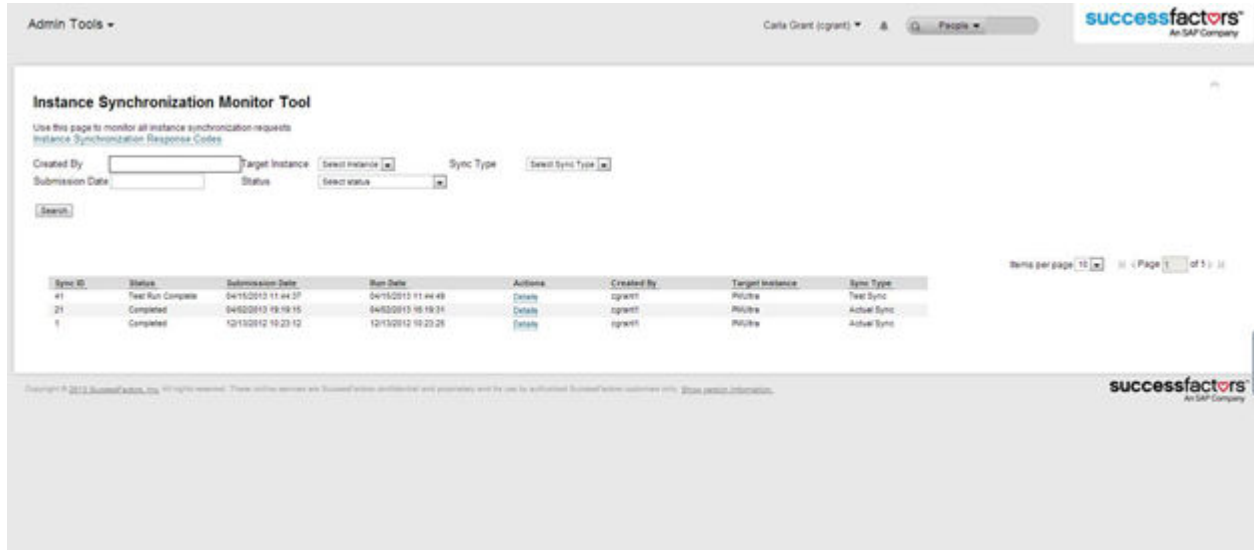
## 18.3 Monitoring a Sync Operation

This section explains how you can monitor your synchronization.

From the admin tools, you can use the instance synchronization monitor tool to display the progress of a sync operation. You can also use the monitor tool to view a summary of the objects that were transferred. All administrators with access to the sync tool have access to the monitor tool.

Navigate to the [Admin Center](#).

In the Tools search field, type *Instance Synchronization Monitor Tool*.



Here is a description of the columns in the monitor tool.

| Column Header   | Description   |
|-----------------|---|
| Sync ID         | ID of the sync request that was run                     |
| Status          | Status of the sync request                              |
| Submission Date | Date of submission                                      |
| Run Date        | Date the request ran                                    |
| Actions         | Details of what the sync request included               |
| Created By      | User ID of the user who created the request             |
| Target Instance | Company ID to which the configurations are being pushed |
| Sync Type       | Test sync or actual sync                                |

## 18.4 Integrating with Instance Sync Tool

This section explains which MDF entities you can sync.

Using the instance sync tool, you can sync the following:

- MDF picklists
- MDF object definitions
- MDF configurable UIs



If you select MDF object definitions and rules, the tool syncs the rules first and then the object definitions. When syncing the rules, the tool does not sync up the trigger point between the object definition and the rule. Note that the trigger point is not the same as the base object attribute within the rule. Each rule has a base object attribute that contains the name of the object on which the rule is based. Then for each object definition, you can mention the rules that are executed and also the event such as *onSave*, *onInit*, or *onFieldChange*. This is called a trigger point, that is, an event upon which a rule is triggered.

When syncing an object definition, you also sync the associations that the object has with any rules. If the corresponding rules are not available in the target, the object definition is not synced. This is then indicated in the Instance Sync audit report.

## Example

The source has an object definition with two rules and the target does not have the object definition and rules. The user selects both the rules and the object definition in the *Instance Sync Wizard* and performs a sync-up. Everything gets synced with the target. Now if the user modifies one of the rules and performs a sync-up again by selecting only the modified rule, the target ends up with the modified rule. But if the user adds a new rule, associates it to the object definition and performs a sync-up by selecting only the new rule and not the object definition, the new rule is synced and created on target. But it is not associated with the object definition.

## Example

The source has an object definition with two rules and the target does not have the object definition and rules. The user wants to sync up the object definition and only one of the rules. So the user selects one of the rules and the object definition when doing the instance sync. The tool syncs up with the rule. But the object definition sync fails since it always tries to sync up all the rules association. The tool cannot find the corresponding rule for one of the associations. So it is not possible to sync object definition and only one of the rules. The *Instance Sync Monitor* logs the information indicating that the object definition was not synced due to the missing rule.

## 18.5 Integrating with SuccessStore

This section explains how you can integrate with the SuccessStore.

You can publish content, such as MDF Custom Object Definitions, Rules, Config UIs, and picklists in the SuccessStore. Using a streamlined process, you can automatically import the specified content into the company instance. A user at any data center can see the content once it is published in SuccessStore.

The screenshot displays the 'SuccessStore Management' interface. At the top, there is a navigation bar with links for 'Company Details', 'New Company', 'Clone Company', 'Reports', 'Operations/Maintenance', 'SMB', 'Manage Provisioners', 'SuccessStore', 'Upgrade Center Media Content', and 'Search Index'. The main content area is divided into two sections: 'Content Basic Information' and 'Content Locale Information'. In the 'Content Basic Information' section, fields include 'Content Id' (7000002), 'Revision' (1.0.0), 'Content Type' (MDF), 'Custom Field' (Position Management), and 'Domain' (shfjshdfgsdf, dfjghjdf). There are also radio buttons for 'Default option in Setup Wizard', 'Display in Setup Wizard', and 'Best Practice option in Setup Wizard'. A blue callout box with the text 'You can view or change the MDF object' points to the 'View File' and 'Change File' links. The 'Content Locale Information' section features a table with columns for 'Locale', 'Name', 'Preview Image', and 'Action'. The table has two rows: one for 'English\_US' with the name 'Second MDF Sample Objects' and another for 'French' with the name 'Second MDF Sample Objects for French'. A red callout box with the text 'Names vary across different locales' points to the 'Name' column. Below the table are 'Save' and 'Cancel' buttons.

You can export an MDF object, such as US Holiday Calendar and store it in SuccessStore. You can then import it to the customer instance. You do not need to export all the data of the MDF object. You can filter data using the external code to select specific object records for export.

### Example

To assign other holidays (another MDF object) to the Holiday Calendar, proceed as follows:

1. Select *US Holiday Calendar* as the parent object.
2. Select *Yes* from the *Include dependencies* dropdown.
3. Click *Export*.

The entire object hierarchy, the holiday calendar as well as holidays, is exported.

## 18.6 Troubleshooting the Instance Sync Tool

This section will help you with some issues related to the Instance Sync Tool.

### Issues with CurrencyConversionRate and CurrencyConversionType MDF Objects

You must not use the MDF objects CurrencyConversionRate and CurrencyConversionType – they are not released for productive usage. If they are in your system, they should be deleted.

1. Navigate to the *Admin Center*. In the *Tools* search field, enter *Configure Object Definitions*.
2. In the search fields, select the *CurrencyConversionType*.
3. Select ► *Take Action* ► *Permanently Delete Entry* ✕.
4. Save your changes.
5. In the search fields, select the *CurrencyConversionRate*.
6. Select ► *Take Action* ► *Permanently Delete Entry* ✕.

### Currency Exchange Rate Syncing between Employee Central and Compensation/Variable Pay

The legacy currency conversion table *ECT\_CONV\_TABLE* has been migrated to the new CurrencyExchangeRate MDF object with currencyExchangeRateType = DEFAULT. For customers who don't use Employee Central, the behavior of the tool for maintaining *ECT\_CONV\_TABLE* will stay the same. Employee Central uses the Currency Exchange Rate object. If you also use Compensation and Variable Pay in your system, then you will have to manually sync the *ECT\_CONV\_TABLE* and Currency Exchange Rate MDF Object.

To check the table:

1. Navigate to the *Admin Center*. In the *Tools* search field, enter *Manage Currency Conversion Rate Tables*.
2. Select the *ECT\_CONV\_TABLE* from the list.
3. When you try to update this table, an information message appears informing you that that table has been migrated. Click *OK*.
4. It is still possible to update this table. Make your changes and save.

For more information, see the Maintaining Currency Exchange Rate GO section of the Migrating to MDF Foundation Objects guide.

# 19 Change History

Learn about changes to the documentation for Implementing the Metadata Framework (MDF) in recent releases.

## 1H 2024

| What's new      | Description  | More Information   |
|-----------------|--|--|
| May 17, 2024    |  |  |
| Added           | Added a note about the actions available for the MDF to-do tasks on the home page.             | <a href="#">MDF Workflows on the Home Page [page 190]</a>    |
| April 12, 2024A |  |  |
| Changed         | Updated information about the order of MDF rule execution.                                     | <a href="#">Order of MDF-Based Rule Execution [page 159]</a> |
| New             | Added information about bulk removal of non-actionable MDF records with invalid workflow data. | <a href="#">Workflows in MDF [page 186]</a>                  |

## 2H 2023

| What's new | Description   | More Information  |
|------------|---|---|
| Changed    | Added information on RTF formatting.                      | <a href="#">Adding Fields [page 25]</a>                                       |
| New        | Added information on supported BB codes.                  | <a href="#">Supported BB Codes for RTF Formatting [page 32]</a>               |
| Changed    | Updated Configuration UI guidelines.                      | <a href="#">Important Configuration Guidelines and Usage Limits [page 10]</a> |
| Changed    | We have moved the Change History to the end of the guide. | <a href="#">What Is the Metadata Framework? [page 7]</a>                      |





# 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 an 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.

## 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.

## 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.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.



© 2025 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.