PUBLIC

MDG-F Overview

Applicable Releases:

All

Version 2.0

Document History

Document Version	Description
2.0	Layout update (Dec 2023)
1.8	Update Layout
1.7	New S/4 HANA MDG 1809 / MDG 9.2 related Information
1.5	Additional Information about Validations
1.4	Updates for Data Transfer including Initial Load, MDG Hierarchies, FAQ
1.3	New Frequently Asked Questions
1.2	Updates for MDG Feature Pack
1.1	Updated broken links and search help information
1.0	First release of the document

1 Business Scenario

SAP Master Data Governance (MDG) provides business processes to find, create, change, and mark master data for deletion. It supports the governance of master data in a central hub and the distribution to connected operational and business intelligence systems.

The processes are workflow-driven and can include several approval and revision phases, and the collaboration of all users participating in the master data maintenance.

MDG offers change request (CR)-based processing of master data with integrated workflow, staging, approval, activation, and distribution.

This guide provides you with the foundation knowledge you need to know about financial data and its related governance solution financial governance (MDG-F).

2 Background Information

With the releases MDG 9.2 and SAP S/4 HANA MDG 1809, MDG-F has different code lines. S/4 HANA MDG 1809 has introduced the simplified maintenance of G/L Account and Cost Elements, similar to the behavior of SAP S/4HANA.

As other content and topics are still identical, this guide applies to both MDG versions on SAP S/4HANA and ERP and indicates specifically whenever it is different.

All software artifacts of the MDG hub that are provided or used by MDG-F belong to the MDG_FND software layer.

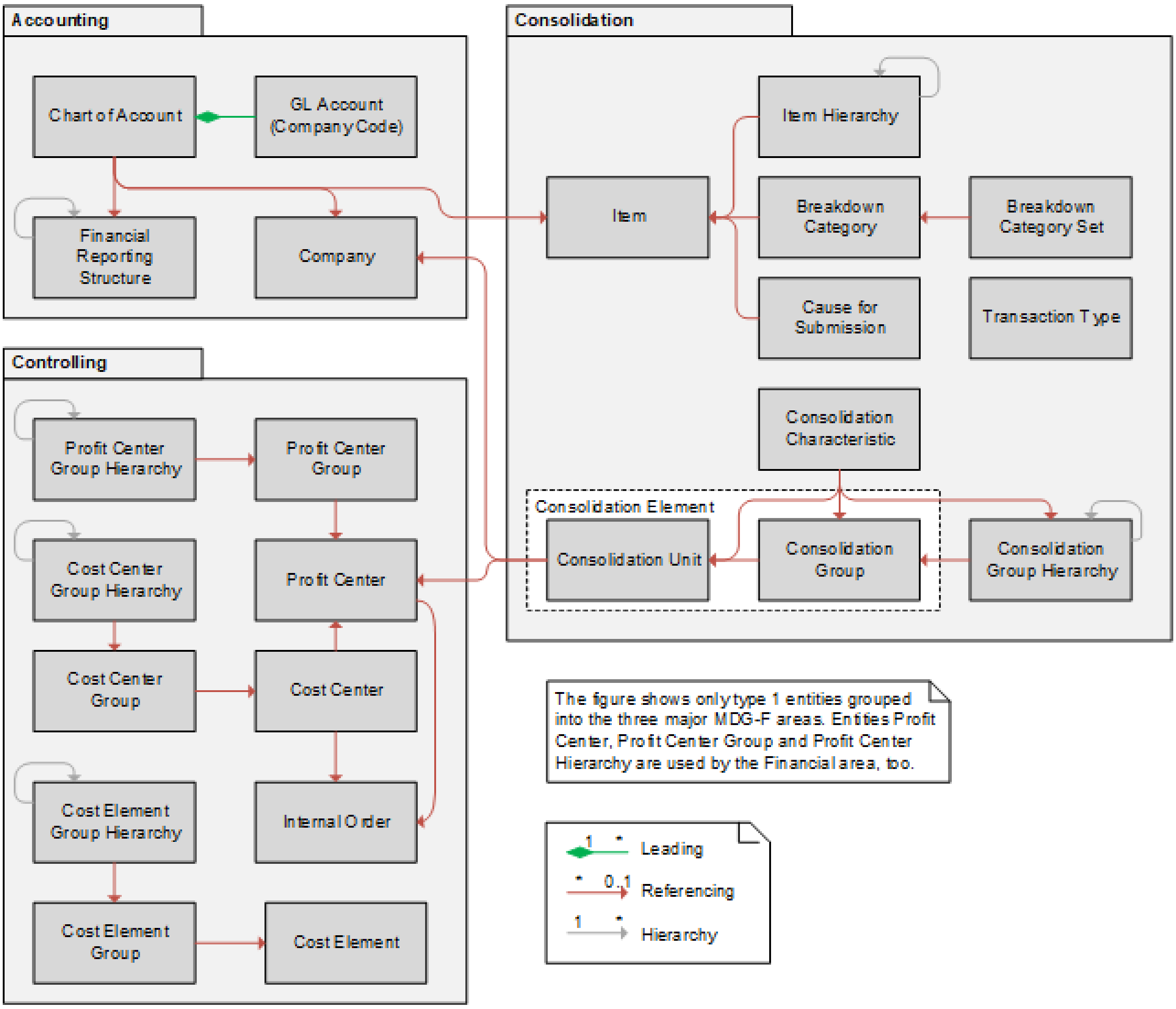
If you have installed MDG on SAP ERP, and when you use either the accounting components or the controlling components of MDG-F, we recommend that the MDG hub includes the SAP_APPL layer. This layer owns customizing tables that the relevant accounting entity types or controlling entity types use.

If you use the consolidation components of MDG-F only, the MDG_APPL and SAP_APPL software layers are optional.

If you have installed MDG on SAP S/4 HANA, all required customizing components are already included.

3 Financial Master Data

MDG offers a single data model combining several entity types of financial master data. The entity types can be separated roughly in three areas: accounting, controlling and consolidation. Nevertheless most of the entities are connected with each other. The figure below shows the currently supported entity types and their relations.



The figure shows only type 1 entities grouped into the three major MDG-F areas. Entities Profit Center, Profit Center Group and Profit Center Hierarchy are used by the Financial area, too.

For an excel file providing a detailed overview of all fields in MDG-F, see SAP note 2021246 MDG Financials – (Un-) Supported Segments and Fields.

3.1 Data Model 0G

The MDG-F entity types, their attributes, and relations are shipped within the data model 0G.

The data model uses only flexible entity types. Both the actual (active) data as well as the staging (inactive) data that exists only within a governance process, is stored in generated MDG tables. Although some entity types refer to existing objects of SAP systems, the active master data is not stored directly in the related backend tables. It requires a data replication to send the MDG active data into the related SAP system tables.

Some of the entity types with SU Type 1 reference other entity types with SU Type 1 as attributes (for example the entity type “account” uses the entity type “company” as an attribute “trading partner”). Using a reference requires that the record to be referenced already exists as an active one within MDG-F:

- Keep the validity of an object according to the chosen edition in mind. If you have created a company that is valid from the 1st of January 2015, the same company cannot be used for an account that is valid from 1st of January 2013 since the company does not exist on the 1st of January 2013.

- Keep the sequence of objects to be created in mind. You must create the company that an account uses before creating the account. It is not sufficient to create the change request for the company only. The change request must be processed completely so that the company record is activated.
- Keep the edition and change request type in mind. If you want to create dependent objects at the same time within a single change request, you have to define both the edition and change request type accordingly. They must contain both entity types.

Each of the MDG-F entity types with SU Type 1 uses the MDG Edition functionality. Editions enable the grouping of different entity types to ensure the data consistency across all entity types. Furthermore, the edition enables the time-dependent maintenance of master data. This is required for some of the related SAP ECC objects like cost or profit centers.

Using editions for the master data maintenance requires that you configure Edition Types to indicate which of the entity types with SU Type 1 can be grouped within an edition. It is recommended to use the SAP pre-defined edition type 0G_ALL. The edition type uses all entity types with SU Type 1. If you want to create your own edition types, keep in mind that:

- Many entity types with SU Type 1 reference other entity types with SU Type 1.
- You cannot assign the same entity type with SU Type 1 to multiple edition types as long as the edition types are not released.
- If objects are not part of the same edition, you will not be able to create cross references.

Additionally Change Request Types are needed for master data maintenance. SAP pre-defines both Edition Types and Change Request Types in release dependent BC-Sets.

Each MDG-F entity type with SU Type 1 uses external and non-changeable keys. The entity’s key must be defined manually during the creation of the entity. It cannot be changed. With MDG 7.0 SP02 MDG-F offers more flexibility with regard to changeable keys during the creation process of an object (see the related how-to guide for more details).

Most of the MDG-F entity types with SU Type 1 support attachments. This enables adding documents and/or links to the master data. The attachments are a MDG specific functionality. They are not related to the SAP ECC document features and thus not part of data replication.

All hierarchies that are defined in data model 0G use the MDG Edition functionality. It is possible to create the same hierarchy in different, multiple points in time. This allows a different structure, for example one for the active business year and a second one for planning the next business year. Note that each hierarchy is an independent object. Following the example above, if you change something in the hierarchy for the active business year, the same change is not applied automatically to the planning hierarchy of the next business year. You have to repeat the change in the planning hierarchy.

In MDG 7.0 it is only possible to maintain hierarchies using the hierarchy processing user interface.

MDG 8.0 introduces the hierarchy maintenance within the single object maintenance user interfaces for accounting-related and controlling-related entity types.

3.2 Accounting

Entity types of the accounting area relate to common SAP ECC objects:

MDG Entity Type	Description
Chart of Accounts (ACCOUNT)	Reflects the so called “A-Segment” of the SAP ECC Account (tables SKA1 and SKAT). It is the chart of accounts dependent data of an account.
G/L Account in Company Code (ACCCCDET)	Reflects the so called B-Segment of the SAP ECC Account (table SKB1). It is the company code dependent data of an account.
Entity types ACCOUNT and ACCCCDET are usually combined within MDG-F to the term G/L Account.	

Company (COMPANY)	Reflects the SAP ECC Company customizing (table T880).
Financial Reporting Structure (FRS)	Reflects the SAP ECC Financial Statement Version / Financial Reporting Structure (table T011).
FRS Item (FRSI)	Reflects the SAP ECC Financial Statement Version / Financial Reporting Structure Items (tables FAGL_011*).
Entity types FRS and FRSI are usually combined within MDG-F to the term Financial Reporting Structure. The financial reporting structure supports the creation of a hierarchy, too. You can assign accounts to this hierarchy.	

3.3 Controlling

Entity types of the controlling area relate to common SAP ECC objects:

MDG Entity Type	**Description**
Cost Center (CCTR)	Reflects the SAP ECC Cost Center (tables CSKS and CSKT).
Cost Center Group (CCTRG) Cost Center Group Hierarchy (CCTRH)	Reflect the SAP ECC Cost Center Group Hierarchy. In MDG-F the entity Cost Center Group Hierarchy defines the root node of the hierarchy. Both entities are required to build the full hierarchy. Cost Centers are added as leafs.
Cost Element (CELEM)	Reflects the SAP ECC Cost Element (tables CSKA, CSKB, and CSKU). In MDG-F a Cost Element does not differentiate between common and controlling area dependent data as done in SAP ECC.
Cost Element Group (CELEMG) Cost Element Group Hierarchy (CELEMH)	Reflect the Cost Element Group Hierarchy. In MDG-F the entity Cost Element Group Hierarchy defines the root node of the hierarchy. Both entities are required to build the full hierarchy. Cost Elements are added as leafs.
Profit Center (PCTR)	Reflects the SAP ECC Profit Center (tables CEPC, CEPC_BUKRS and CEPCT).
Profit Center Group (PCTRG) Profit Center Group Hierarchy (PCTRH)	Reflect the Profit Center Group Hierarchy. In MDG-F the entity Profit Center Group Hierarchy defines the root node of the hierarchy. Both entities are required to build the full hierarchy. Profit Centers are added as leafs.
Internal Order (IORDER)	Reflects the SAP ECC Internal Order (tables AUFK).

3.4 Consolidation

Entity types of the consolidation area relate to the SAP solution for Business Consolidation Services of Strategic Enterprise Management (SAP SEM BCS):

MDG Entity Type	Description
Consolidation Characteristic (CONSCHAR)	Defines the basis for consolidation groups and consolidation elements. The characteristic’s scope either relates to a company or a profit center.
Consolidation Unit (CONSUNIT)	Defines the consolidation unit of a consolidation element. Depending on the used consolidation characteristics, either company or profit center specific data can be maintained.
Consolidation Group (CONSGRP)	Defines the consolidation group of a consolidation element.
Entity types CONSUNIT and CONSGRP are usually combined within SAP SEM BCS Consolidation Element. In MDG-F both are treated as separated entity types.	
Consolidation Group Hierarchy (CONSGRPH)	Defines the root node of the consolidation group hierarchy. Consolidation groups and units can be added to the hierarchy.
Item (FSI)	Defines the item. MDG-F uses Items for the maintenance of Group Accounts, too. Therefore entity type ACCOUNT has a cross reference to FSI.
Item Hierarchy (FSIH & FSIT)	Reflect the item hierarchy resp. consolidation reporting structure of SAP SEM BCS. Entity type FSIH defines the root node of the hierarchy. Entity type FSIT defines text items usable as sub-nodes of the hierarchy. Item can be assigned as leafs.
Breakdown Category (BDC)	Defines the breakdown category. Each category uses sets. A category is usually assigned to an item.
Breakdown Category Set (BDCSET)	Defines the breakdown category sets.
Cause for Submission (SUBMPACK)	The cause for submission does not exist in SAP SEM BCS. The entity type serves as an example how-to extend MDG-F with additional information.
Transaction Type (TRANSTYPE)	Defines the transaction type.

All entity types are created according to the SOFEX example model that is shipped as a template model within SAP SEM BCS. Any changes performed in the SAP SEM BCS model have to be applied to MDG-F, too.

4 User Interface

The user interfaces for MDG-F are based on ABAP Web Dynpro. They are built with the floor plan manager (FPM) using its specific Business Object Layer (BOL) / Generic Integration Layer (genIL) technology. Some advantages to be mentioned are:

- Loose coupling of the user interfaces to the MDG-specific processes.

- High flexibility for the creation of the user interfaces. The huge amount of fields to be displayed is split into small User Interface Building Blocks (UIBBs). UIBBs support lists, forms and special kinds like pop-ups, search input and search results.
- Possibility to create object-specific user interfaces to create a common look and feel and/or a similarity of the MDG user interfaces compared to the SAPGUI maintenance transactions.
- Reuse of the MDG-F generated tables, structures and fields (including naming) during the user interface creation.
-

General information about FPM and its functionality is available in the FPM Cookbook.

4.1 Generic Interaction Layer (genIL)

The generic interaction layer (genIL) is required for the MDG-F user interfaces. It consists of a genIL model and one or more genIL implementation classes for the specific model. Transaction GENIL_MODEL_BROWSER can be called in the SAP backend to view the model. A genIL model basically consists of objects and relations.

- Objects consist of attributes. Each attribute reflects a usable field for the user interface.
- Relations connect one object to another. They define the cardinality of objects in a relation, too. Relations are reflected in the user interface by the wires (connections) from one UIBB to another. It is mandatory that the UIBB hierarchy in the overview page is consistent to the genIL object hierarchy as defined by the relations.

SAP provides the following genIL models for MDG-F:

- MDG 7.0: Model MDGF and its implementation class CL_MDGF_GENIL_ADAPTER.
- MDG 8.0: Model MDGFHP and its implementation class CL_MDGF_GENIL_ADAPTER_HRY.

Both models are dynamic genIL models. It is strictly forbidden to change the models manually. The models are generated by their implementation classes according to the given run-time information of the MDG-F data model 0G.

Model MDGF is used by the single object maintenance user interface for the common entity types and their attributes.

- Each entity type with SU Type 1 of the data model is transferred to a genIL root object.
- Each entity type with SU Type 4 of the data model is transferred to a genIL dependent object.
- Relations (for example, between entity types with SU Type 1 and other entity types with SU Type 1; or between entity types with SU Type 1 and entity types with SU Type 4) are determined and transferred into genIL relations.
- Each entity type with SU Type 1 retrieves additional genIL query and query result objects to support the search.
- If an entity type with SU Type 1 supports multi-lingual texts, a dependent object is created in genIL to enable the text maintenance within a table.
- If an entity type with SU Type 1 supports attachments, two dependent objects are created in genIL to enable the attachment maintenance within a table and related pop-ups.
- The generated structures belonging to an entity are used for the genIL key and attribute structures. This ensures that all fields of the MDG data model 0G are available for the creation of the related user interfaces. Attribute structures are used by FPM to build the field catalog being available during the user interface creation.

Model MDGFHP is used by the single object maintenance user interface for the entity types that are used within hierarchies.

- Each entity type with SU Type 1 that is used by hierarchies as a root node is transferred into a genIL root object. Those objects are used to build the hierarchy tree in the user interface.
- Each entity type with SU Type 1 that defines a hierarchy is transferred into a genIL dependent object. Those objects are used to build the hierarchy tree in the user interface.

- Each entity type with SU Type 1 that is used by hierarchies as a root node, a grouping node, or leaf node is transferred into a genIL access object. Those objects are used to build the hierarchy assignment lists in the user interface.
- Each entity type with SU Type 1 that is used by hierarchies as a root node, a grouping node, or a leaf node retrieves an additional genIL query objects to support the search. The search result is displayed as hierarchy tree.
- If a hierarchy defines hierarchy attributes for a specific entity type with SU Type 1, a corresponding genIL dependent object is created additionally.

Caution

Enhancements of the MDG-F data model 0G are reflected immediately after activation of the data model in the genIL components. A manual change or enhancement of the genIL components is strictly forbidden.

If enhancements in genIL are required, all changes have to be implemented in a related genIL implementation class. It is mandatory that this class inherits from the SAP classes CL_MDGF_GENIL_ADAPTER respectively CL_MDGF_GENIL_ADAPTER_HRY.

4.2 Context Based Adaptations

A context based adaptation (CBA) is an FPM concept that allows changing the user interface in a flexible way based upon given values (for example application parameters, user input, and others). A CBA consists of an Adaptation Schema that consists of one or more Adaptation Dimensions.

- The pre-defined adaptation schema for MDG-F is MDG_FIN.
- The adaptation schema includes the following adaptation dimensions:
    - USMD_OTC: usable for adaptations according to the current business object type code.
    - ACTION: usable for adaptations according to the current logical action like “create” or “mark for deletion”.
    - CRTYPE: usable for adaptations according to the current change request type.
    - WFSTEP: usable for adaptations according to the current workflow step.

Using both the adaptation schema and its dimensions, you can create various adaptations of the user interface (for example changing the layout of an overview page (OVP), or either adding rows to or removing rows from a list UIBB, and so on). You can combine several dimensions to create a very specific adaptation.

The CBA concept is based on the common FPM event handling. You can trigger one or more CBA events that are handled by FPM's event loop processing. Unfortunately FPM handles multiple CBA events one after the other. It does not cumulate the dimension information given within each event (which basically means that the final CBA event wins).

Additional information is available in the FPM Cookbook in chapter Context Based Adaptations (CBA).

4.3 Work Centers, Roles and Landing Pages

Work centers and roles are used within MDG for the creation of so called landing pages. A landing page is considered as a single point of entry for an end user to the MDG-F specific user interface.

Work centers define a grouping of the different MDG-F entity types into a single menu role. Additionally each role uses the common MDG components such as data replication, change request, editions, and so on. Three work center menu roles are pre-defined according to the MDG-F data model setup:

- SAP_MDGF_ACC_MENU_*: reflects all accounting entity types.
- SAP_MDGF_CTR_MENU_*: reflects all controlling entity types.

  - For SAP S/4 HANA MDG 1809 and higher releases, after the business function MDG_S4_FINANCIALS_8 is activated, you can use SAP_MDGF_CTR_MENU_SFIN_* to access all controlling entity types
- SAP_MDGF_CO_MENU_*: reflects all consolidation entity types.

The * counter varies depending on the actual release you are using.

Next to the menu roles as mentioned above, authorization roles are pre-defined, too. The authorization roles follow the general role setup of all MDG based applications:

- The Display Role contains all authorizations to display master data, change requests, change documents, replication status, and other information for the respective work center data.
- The role is assigned to an auditor, who is then authorized to display all information, but not to change anything.
- The Requester Role contains all authorizations to create and display change requests within the respective work center. The role is assigned to a business user, who is then able to request changes for master data.
- The Specialist Role contains all authorizations to process and approve change requests in the respective work center.
- The Steward Role contains all authorizations needed for special tasks that involve the processing and replication of master data within the respective work center. The role is assigned to the user responsible for the quality of the master data. It is recommended to combine this role with the specialist.

The combination of the work center and the explained general authorization roles results in a set pre-defined roles:

Role	Description
SAP_MDGF_ACC_DISP_*	MDG-F: Accounting Display
SAP_MDGF_ACC_REQ_*	MDG-F: Accounting Requester
SAP_MDGF_ACC_SPEC_*	MDG-F: Accounting Specialist
SAP_MDGF_ACC_STEW_*	MDG-F: Accounting Data Steward
SAP_MDGF_CTR_DISP_*	MDG-F: Controlling Display
SAP_MDGF_CTR_REQ_*	MDG-F: Controlling Requester
SAP_MDGF_CTR_SPEC_*	MDG-F: Controlling Specialist
SAP_MDGF_CTR_STEW_*	MDG-F: Controlling Data Steward
SAP_MDGF_CO_DISP_*	MDG-F: Consolidation Display
SAP_MDGF_CO_REQ_*	MDG-F: Consolidation Requester
SAP_MDGF_CO_SPEC_*	MDG-F: Consolidation Specialist
SAP_MDGF_CO_STEW_*	MDG-F: Consolidation Data Steward

Each pre-defined role must be considered as a template. The roles must not be assigned directly to user. An SAP template role must be copied and adjusted according to the customer project requirements.

4.4 MDG-F User Interfaces

All software artifacts related to the new MDG-F user interfaces for single object maintenance are stored in package USMDZ10.

A complete user interface consists of the following elements:

- The application configuration that defines the general settings of a user interface including application parameters.
- The communicator settings configuration that defines the search of each overview page. The communicator settings configuration must have the same name as the application configuration.
- The Overview Page (OVP) that combines one or more form UIBBs and/or list UIBBs for the actual data maintenance. Forms, lists and their related feeder classes can be identified by their naming.
    - Web Dynpro Component MDGF_0G_CCTR belongs to the cost center.
    - Feeder class CL_MDGF_GUIBB_FI_ACCOUNT belongs to the account.

The new user interface uses the business object type code (OTC) as a key parameter. The OTC links the data model and the entity type with the user interface application. It is mandatory that each user interface uses an OTC. Pre-defined combinations are listed in the next chapters.

4.4.1 Accounting

Application / OVP / OTC	Entity Type(s)	Description
MDGF_0G_OVP_FI_ACCOUNT MDGF_0G_FI_ACCOUNT_OVP 892	ACCOUNT ACCCCDET	The OVP combines two entity types with SU Type 1 for account master data maintenance in a single user interface. Accounts can be assigned to financial reporting structure hierarchies in collective processing.
MDGF_0G_OVP_COMPANY MDGF_0G_COMPANY_OVP 154	COMPANY	The OVP represents the company master data.
MDGF_0G_OVP_FI_REPORT MDGF_0G_FI_REPORT_OVP 901	FRS FRSI	The OVP combines two entity types with SU Type 1 for financial reporting structure maintenance in a single user interface. The user interface creates single entities that can be used to define the complete financial reporting structure hierarchy in collective processing.

4.4.2 Controlling

Application / OVP / OTC	Entity Type(s)	Description
MDGF_0G_OVP_CCTR MDGF_0G_CCTR_OVP 158	CCTR	The OVP represents the cost center master data. The assignment of the cost center to its (standard) hierarchy must be done in collective processing.

MDGF_0G_OVP_CCTRG MDGF_0G_CCTRG_OVP 895	CCTRG	The OVP represents cost center groups. The assignment of the cost center groups to hierarchy nodes must be done in collective processing.
MDGF_0G_OVP_CCTRH MDGF_0G_CCTRH_OVP 897	CCTRH	The OVP represents the root node of a cost center group hierarchy. Hierarchy maintenance must be done in collective processing.
MDGF_0G_OVP_CELEM MDGF_0G_CELEM_OVP 983	CELEM	The OVP represents the cost element master data. The assignment of the cost center to its hierarchy must be done in collective processing.
MDGF_0G_OVP_CELEMG MDGF_0G_CELEMG_OVP 984	CELEMG	The OVP represents cost element groups. The assignment of the cost element groups to hierarchy nodes must be done in collective processing.
MDGF_0G_OVP_CELEMH MDGF_0G_CELEMH_OVP 985	CELEMH	The OVP represents the root node of a cost element group hierarchy. Hierarchy maintenance must be done in collective processing.
MDGF_0G_OVP_PCTR MDGF_0G_PCTR_OVP 229	PCTR	The OVP represents the profit center master data. The assignment of the profit center to its (standard) hierarchy must be done in collective processing.
MDGF_0G_OVP_PCTRG MDGF_0G_PCTRG_OVP 896	PCTRG	The OVP represents profit center groups. The assignment of the profit center groups to hierarchy nodes must be done in collective processing.
MDGF_0G_OVP_PCTRH MDGF_0G_PCTRH_OVP 898	PCTRH	The OVP represents the root node of a profit center group hierarchy. Hierarchy maintenance must be done in collective processing.
MDGF_0G_OVP_IORDER MDGF_0G_IORDER_OVP DRF_0019	IORDER	The OVP represents the internal order master data.

4.4.3 Consolidation

Application / OVP / OTC	Entity Type(s)	Description
MDGF_0G_OVP_BDC MDGF_0G_BDC_OVP MDGF_BCG	BDC	The OVP represents the breakdown category master data. Breakdown categories include breakdown category sets. They can be used to refine items.
MDGF_0G_OVP_BDCSET MDGF_0G_BDCSET_OVP MDGF_BCS	BDCSET	The OVP represents breakdown category sets.
MDGF_0G_OVP_CONSCHAR MDGF_0G_CONSCHAR_OVP MDGF_CCC	CONSCHAR	The OVP represents the consolidation characteristic master data.
MDGF_0G_OVP_CONSGRP MDGF_0G_CONSGRP_OVP 904	CONSGRP	The OVP represents the consolidation group. The assignment of the consolidation groups to hierarchy nodes must be done in collective processing.
MDGF_0G_OVP_CONSGRP MDGF_0G_CONSGRP_OVP 894	CONSGRPH	The OVP represents the root node of a consolidation group hierarchy. Hierarchy maintenance must be done in collective processing.
MDGF_0G_OVP_CONSUNIT MDGF_0G_CONSUNIT_OVP 905	CONSUNIT	The OVP represents the consolidation unit master data. Consolidation units can be assigned to consolidation group hierarchies in collective processing.
MDGF_0G_OVP_CO_ACCOUNT MDGF_0G_CO_ACCOUNT_OVP 900	FSI	The OVP represent the item. Items can be assigned to item hierarchies in collective processing.
MDGF_0G_OVP_CO_REPORT MDGF_0G_CO_REPORT_OVP 902	FSIH FSIT	The OVP combines two entity types with SU Type 1 for item hierarchy maintenance in a single user interface. The user interface creates single entities that can be used to define the complete item hierarchy in collective processing.
MDGF_0G_OVP_SUBMPACK MDGF_0G_SUBMPACK_OVP	SUBMPACK	The OVP represents the cause for submission. A cause for submission is used to refine items.

MDGF_SMP		
MDGF_0G_OVP_TRANSTYPE MDGF_0G_TRANSTYPE_OVP MDGF_TTP	TRANSTYPE	The OVP represents the transaction types.

4.4.4 User Interfaces for Collective Processing

The previous chapters Accounting, Controlling and Consolidation listed user interfaces for single object maintenance.

Collective processing summarizes the user interfaces for both mass change and for hierarchy maintenance. The related user interfaces are not refactored. They are accessible through links in the related work centers.

4.4.5 Search and Value Helps for Single Object Maintenance User Interfaces

MDG-F uses the generic text helper functionality provided by the MDG framework with class CL_USMD_GENERIC_GENIL_TEXT. The text helper is able to provide values of domains and check tables that have only a single key field. This functionality is combined with the OVS search technology of the user interface:

Fields having domain fixed values are usually displayed as drop-down list boxes in the user interface. The values of the drop-down list box are determined by the text helper.

Fields having simple check tables are usually displayed as input fields including a related read-only field for the description. MDG-F uses the generic OVS search implementation provided by FPM (see class CL_MDGF_GUIBB_FORM). The text helper is called indirectly by the MDG-F specific genIL implementation.

Fields having complex check tables or depending on other values are usually displayed as input fields including a related read-only field for the description. MDG-F implements specific OVS search helps in the related feeder classes since this cannot be done with a generic approach.

There are two options for filtering search help results:

The text helper includes the enhancement spot USMD_GENERIC_GENIL_TEXT respectively BAdI USMD_GENERIC_GENIL_TEXT_BADI. They are usable to restrict the values being shown on the user interface for domains or simple check tables.

The generic form feeder class CL_MDGF_GUIBB_FORM provides a rather generic OVS filter method called OVS_OUTPUT_FILTER. You can enhance this logic (for example, using an implicit enhancement at the end of the method) with your custom filters.

5 Data Transfer

Data Transfer in MDG-F covers both the initial load of master data into and MDG system and the replication of created or modified data to client systems.

The initial load is used for preparing the MDG system with the currently existing data of client systems. The initial load can be accomplished using a file-based approach or using the MDM Generic Extractor. When setting up the system it is important to load all objects that shall be put under governance completely. For time-dependent objects such as Profit Centers or Cost Centers, it is ensured that all existing time slices of a single object are loaded to MDG.

Data replication is possible using either SOA Services or ALE IDocs. Combining both for the same receiving system is not recommended. IDocs are only available for accounting and controlling entity types. Consolidation entity types do not exist within SAP ECC systems and thus do not offer IDocs.

SAP Note (Un-) Supported Segments & Fields offers a detailed overview about all fields covered by MDG-F and how they are used in the different replication technologies.

Data replication supports different modes that are explained in the next chapters.

5.1 Object-based Replication

This is a new feature provided by MDG 7.0. It enables the immediate replication of an existing object, for example by selecting the same in the search result list and choosing the Replicate button. It is possible to select one or more target systems for the current replication run. The edition cannot be selected. The outbound implementation always uses the currently valid data for non-time-dependent objects respectively all valid time slice for time-dependent objects. It is not possible to replicate outdated objects.

5.2 Change request-based Replication

The change-request based replication has not been changed. It is triggered once the change request has been activated. If the changed object is non-time-dependent and the change happened for a time slice that is not valid (for example, in an outdated or future edition), the object is not replicated. This prevents overwriting non-time-dependent objects with invalid data in the target systems.

5.3 Edition-based Replication

The edition-based replication has not been changed. It still requires a released edition to replicate all objects of the same edition. For non-time-dependent objects the same rules as defined by the change-request based replication apply. The system prevents the replication of invalid time slices.

5.4 Replication of Deletions

MDG-F supports the replication of deletion only using SOA messages. It is not possible to use ALE IDocs. A “Deletion” means both the actual deletion of a complete object and the deletion of a single time slice to set the “valid to date” to a specific value.

5.5 Replication of Accounts

MDG-F provides two entity types with SUT1 for the account.

- The entity type ACCOUNT represents the so-called A-Segment of an account. It relates to the SAP ECC table SKA1 Accounts.
- The entity type ACCCCDET represents the so-called B-Segment of an account. It relates to the SAP ECC table SKB1 Accounts in Company Code.

The predefined replication models for the account reflect this setup.

The IDoc based replication uses the common GLMAST IDoc type that combines both the A and B Segments. The replication is triggered using the business object type for the general ledger accounts master (892). It is possible to replicate A segments without B segments. If the object list to be replicated contains only B segments, related A segments are added automatically.

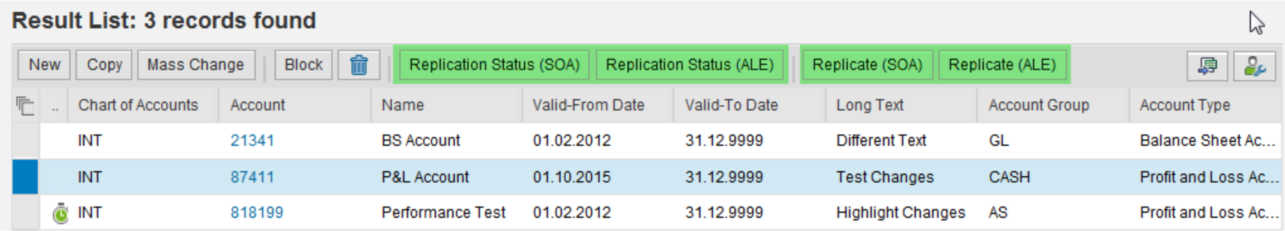
The SOA service-based replication is more complex since there are two separate service messages used for the replication.

- Service ChartOfAccountsReplicationRequest_V1 contains only the A segment without any attributes of the B segment. The replication is triggered using the object type code for the financial accounting chart of accounts (899).

- Service GeneralLedgerAccountMasterReplicationBulkRequest contains additional attributes of the A Segment and the complete list of attributes of the B segment. The replication is triggered using the object type code for the general ledger accounts master (892).

The service definition requires that any replication of A segments must send both of the abovementioned services to ensure data consistency in the receiving system. This is implemented by the corresponding outbound implementations. B segment replication is possible without the chart of accounts service.

A side-effect of the different outbound implementations that are required to ensure data consistency is the design of the search result list for accounts as shown in the screenshot below.



..	Chart of Accounts	Account	Name	Valid-From Date	Valid-To Date	Long Text	Account Group	Account Type
	INT	21341	BS Account	01.02.2012	31.12.9999	Different Text	GL	Balance Sheet Ac...
	INT	87411	P&L Account	01.10.2015	31.12.9999	Test Changes	CASH	Profit and Loss Ac...
⏱	INT	818199	Performance Test	01.02.2012	31.12.9999	Highlight Changes	AS	Profit and Loss Ac...

The button Replication Status (SOA) shows the status for any replication of accounts using the chart of accounts replication request. The replication itself is triggered with the button Replicate (SOA).

The button Replication Status (ALE) shows both the status for any IDoc-based replication of accounts and the status for the SOA-based replication using the general ledger account master service. The replication itself is triggered with the button Replicate (ALE). You choose this button for IDoc based replication only.

5.6 Replication Monitoring

Data replication offers several monitors to check the current replication status.

- The search result list of MDG-F user interfaces offers a link to the DRF replication status. The button Replication Status is active for all entity types that support data replication in general.
- ALE IDoc replication offers a specific monitor. You can use backend transaction WE02 to display both outbound IDocs (on the MDG hub) as well as inbound IDocs (on the MDG client). You can use the monitor for detailed troubleshooting. Post-processing of IDocs is possible using transaction BD87.
- SOA Service replication offers a specific monitor. You can use backend transaction SXMB_MONI (available until SAP Basis 7.32) to display outbound services (on the MDG hub). You can use backend transaction and SRT_MONI (available in SAP Basis 7.40 and newer) to display inbound services (on the MDG client). You can use the monitor for detailed troubleshooting. In the forward error handling, you can post-process services, using backend transaction /SAPPO/PPO2.

6 Validations and Data Derivation

As mentioned before, the data model 0G uses only flexible entity types. This means that there is no access class available that could be used either for validations or for data derivation, or for both. Nevertheless validations and data derivation are needed by the MDG-F entities, too.

MDG-F implements the common MDG rule service BAdI USMD_RULE_SERVICE for both validations and derivations. All software artifacts are stored in package USMDZ7.

You can enhance the existing validations with custom checks, too. The recommended approach is a custom class that inherits from the pre-defined SAP class to ensure, that all SAP pre-defined validations are executed. SAP Note 2337685 “MDG-F: Standard Validations of Data Model 0G” provides a detailed overview of all pre-defined validations in MDG-F.

MDG supports different types of derivation:

- BRF+ based derivation

- Single entity derivation of the rule service BAdI USMD_RULE_SERVICE
- Cross entity derivation of the rule service BADI USMD_RULE_SERVICE_CROSS_ET

The default implementations provided by SAP use only the cross-entity derivation. There is a specific how to guide “Cross Entity Derivation in MDG-F” available that explains all details.

7 FAQ – Data Modelling

7.1 Q: Why is there an active data model 0F in the MDG hub?

Data model 0F has been introduced with the very first shipment of MDG-F for SAP ECC 6.0 EhP4. As of today, the data model is outdated and must not be used anymore. Delete the data model in your system.

7.2 Is it recommended to copy data model 0G?

It is not recommended to create a copy of the data model 0G.

The existing data model can be easily enhanced in various ways:

- Adding new fields to an existing entity type.
- Adding new tables as type 4 entity to an existing entity type.
- Adding new entity types with SU Type 1 to the data model.

The generic implementation of the genIL layer simplifies the integration of the enhancements in the user interface.

7.3 How do I add custom Data Dictionary search helps to attributes of data model 0G?

The new user interfaces for MDG-F consider the following data dictionary-based search helps:

- Search helps that are defined within the related data element or domain.
- Search helps that are defined within the UI configuration, for example using a re-definition of the feeder class method GET_DEFINITION. The UI configuration overrules a definition on data element or domain level.

The new user interfaces ignore all data dictionary-based search help that are maintained in the MDG data model customizing views. It is recommended to use the UI configuration instead.

8 FAQ – User Interface Design

8.1 Is it recommended to copy existing user interfaces?

It is not recommended to copy existing user interfaces.

The existing user interfaces can be easily enhanced in various ways without the need of a copy or a modification of the pre-defined user interfaces:

- You can customize the existing user interface by adding fields or removing fields.
- You can adapt the user interface adapted dynamically using context-based adaptations.
- You can enhance the user interface with custom UIBBs using the enhancement functionality of FPM.

How-to guide “Extending the Data Model by New Fields in MDG-F” provides a real-life example how the user interface can be enhanced without copying.

A copy might destroy the binding to the improvements and corrections made by SAP.

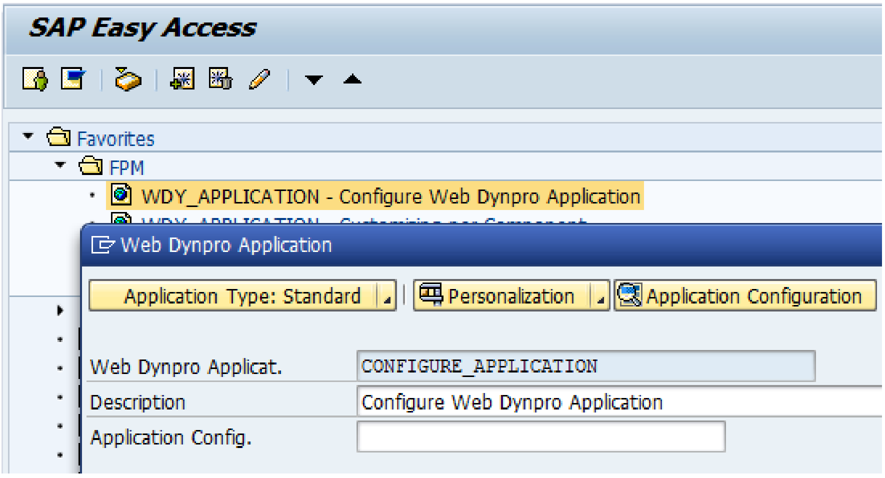
If, in spite of the recommendation, you copy the user interface anyway, ensure that you copy it completely. Remember that each user interface consists of an application configuration, a UI configuration and communicator settings.

8.2 How can I start the user Web Dynpro application configuration for user interfaces?

You use the Web Dynpro application configurator to define your Web Dynpro based applications. It defines the main components of the user interfaces (for example the overview pages to be used) as well as application specific parameters.

The most efficient way to start the Web Dynpro application configurator for user interfaces is using a favorite in the SAP Menu.

1. Logon to your MDG Hub system.
2. Create a new entry in you SAP Menu Favorite List.
3. Choose Web Dynpro Application.
4. Define the Web Dynpro Application as CONFIGURE_APPLICATION.
5. Define any Description.


SAP Easy Access
Favorites
FPM
WDY_APPLICATION - Configure Web Dynpro Application
Web Dynpro Application
Application Type: Standard
Personalization
Application Configuration

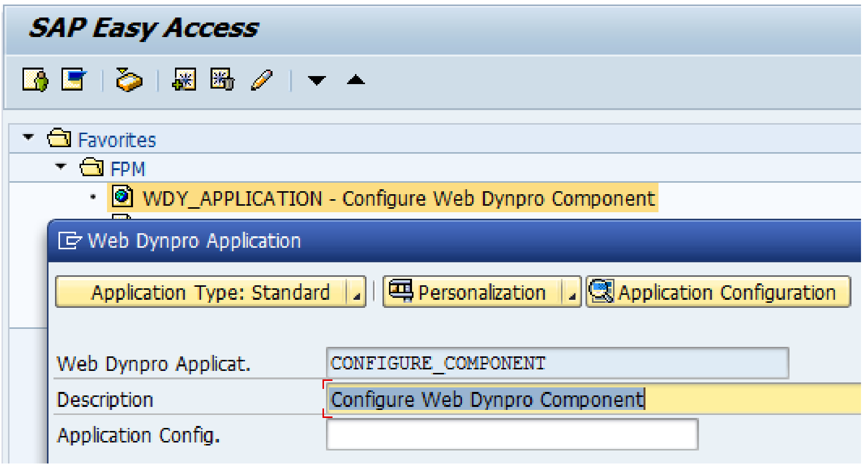
Web Dynpro Applicat. CONFIGURE_APPLICATION
Description Configure Web Dynpro Application
Application Config.

8.3 How can I start the user Web Dynpro component configurator for user interfaces?

You use the Web Dynpro component configurator to create your own Web Dynpro components, for example for MDG Communicator settings.

The most efficient way to start the Web Dynpro component configurator for user interfaces is using a favorite in the SAP Menu.

1. Logon to your MDG Hub system.
2. Create a new entry in you SAP Menu Favorite List.
3. Choose Web Dynpro Application.
4. Define the Web Dynpro Application as CONFIGURE_COMPONENT.
5. Define any Description.


SAP Easy Access
Favorites
FPM
WDY_APPLICATION - Configure Web Dynpro Component
Web Dynpro Application
Application Type: Standard
Personalization
Application Configuration

Web Dynpro Applicat. CONFIGURE_COMPONENT

Description Configure Web Dynpro Component

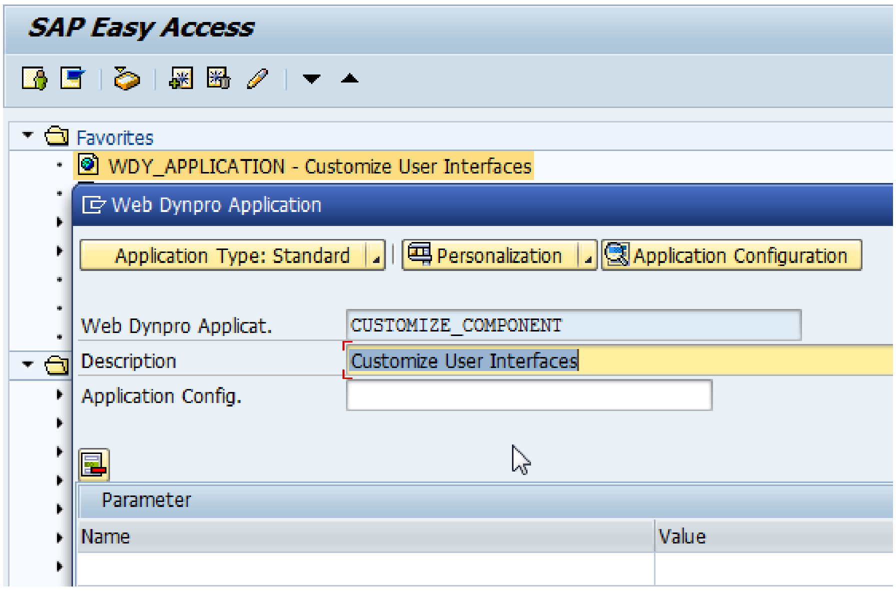
Application Config.

8.4 How can I start the user Web Dynpro customizing configurator for user interfaces?

You use the Web Dynpro component customizing configurator to customize SAP defined user interface components like complete overview pages or single forms or lists.

There are different ways to start the Web Dynpro component customizing configurator for user interfaces:

1. Logon to your MDG Hub system.
2. Create a new entry in you SAP Menu Favorite List.
3. Choose Web Dynpro Application.
4. Define the Web Dynpro Application as CUSTOMIZE_COMPONENT.
5. Define any Description.


SAP Easy Access
Favorites
WDY_APPLICATION - Customize User Interfaces
Web Dynpro Application
Application Type: Standard
Personalization
Application Configuration
Parameter

Web Dynpro Applicat. CUSTOMIZE_COMPONENT

Description Customize User Interfaces

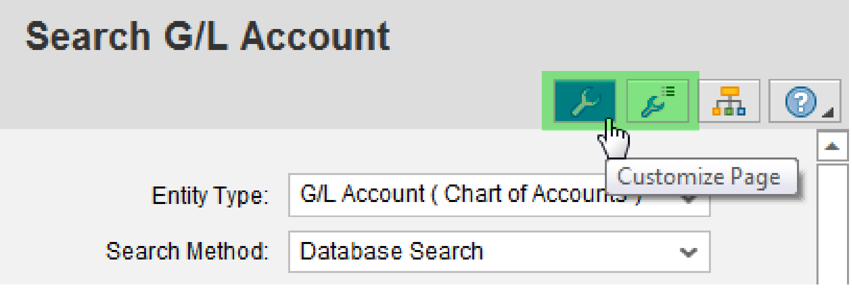
Application Config.

Name	Value

Configure the own user as FPM Configuration Expert

1. Logon to your MDG Hub system.
2. In the top-menu bar choose System à User Profile à Own Data.
3. Switch to tab Parameters.
4. Enter Parameter ID FPM_CONFIG_EXPERT and set its value to X.
5. Save the changes.

If you now start any user interface that is based upon FPM, you'll notice two new buttons in the upper right area of the user interface. You can use both buttons to start the customizing configurator. The screenshot below uses the search for account as example.



Search G/L Account

Entity Type: G/L Account (Chart of Accounts)

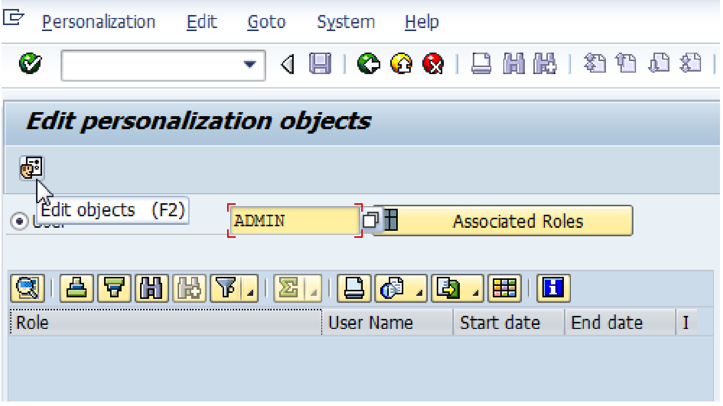
Search Method: Database Search

8.5 Why is there a short dump when I try to customize a user interface?

Before you can create or modify user interface customizing, you must define the Standard Data Model in your user personalization. The model must be set according to the user interface that you want to customize. Ensure that the value is set to 0G for MDG-F related user interfaces:

Start transaction SPERS_MAINT.

Enter your user name and choose Edit objects (F2) above your user name.



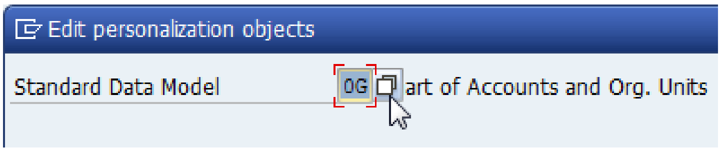
Edit personalization objects

User ADMIN

Use the search with search term MDM to locate the entry SAP Master Data Governance – R_FMDM_MODEL in the table of personalization objects.

Double-click on the entry.

In the pop-up Edit personalization objects set the Standard Data Model to 0G.



Edit personalization objects

Standard Data Model 0G art of Accounts and Org. Units

Save your changes.

8.6 What does the error message “Feeder class does not provide a valid field catalog” mean?

This error message might be displayed if you try to open the web dynpro component configuration for any UIBB. The root cause is the same as for question “Why is there a short dump when I try to customize a user interface?”. The SAP New Weaver Basis release 7.40 displays the error message instead of causing a short dump. To resolve the issue you need to define the Standard Data Model as 0G as described above.

8.7 Why is there still a dump/error message although I have defined the Standard Data Model as 0G?

Technically the assignment of the standard data model to your user is stored in data base table USMD2000. You could either check by yourself or ask your system administrator to check, if there is a valid entry for your user in the table.

In systems that have been upgraded from an older MDG-F release to MDG-F 7.0 (or newer), it might be possible that:

Your user is still assigned to the old MDG-F roles.

The old MDG-F roles define a standard data model via customizing table USMD2001.

If this is the case, the standard data model being displayed by transaction SPERS_MAINT is determined according to your roles. This does not work for the user interface configuration of MDG-F 7.0. The invalid customizing settings (especially, the role assignment for your user to outdated MDG-F roles) have to be deleted. Afterwards you can define the standard data model.

8.8 Why does the context-based adaptation (CBA) not change the layout of the overview page (OVP)?

You can only use CBAs to change the layout of an OVP during the startup of the application. It is not possible to change the OVP (for example, the sequence of UIBBs) using a CBA during UI round-trips. CBAs can only change the layout of single UIBBs for each round-trip.

8.9 Why are recent changes applied to the user interface not visible for other users?

The actual UI that is being displayed to a user in the web browser is determined from various components of the UI configuration:

1. Personalization
2. Enhancements
3. Context Based Adaptations
4. Base Configuration

The general rule is that the personalization is the strongest component. This is best explained with an example:

The base configuration defines the overview page as a list of UIBBs. Since a user does not want to scroll, he or she creates a personalization of the page introducing a stacking of the UIBBs in tab-strips. A UI designer decides to create a context-based adaptation that sets a single UIBB to “hidden and excluded from event loop”. All users not having a personalization no longer see this UIBB anymore. The user with the personalization set is unaffected by this change. This is because the UIBBs that are hidden and excluded from event loop still belong to the OVP. They can be added to the OVP using personalization. Since the user has created a personalization that shows the UIBB (the personalization was created before the CBA), the UIBB is still visible. To exclude the UIBB, you must either reset personalization or delete the UIBB in the CBA.

System administrators might use the ABAP Web Dynpro application WD_ANALYZE_CONFIG_USER to check and/or reset user personalization centrally.

8.10 How do I create a new User Interface for a custom Entity with Storage and Use Type 1?

If you have enhanced the standard data model 0G with a custom entity type with SUT 1, you need to create a new user interface, too. Follow the steps mentioned in the SAP Help for Creating User Interfaces for Single Object Processing. Of course, you can use any of the existing MDG-F user interfaces as template.

9 FAQ – Data Maintenance

9.1 What does the error message “Relation <ABC> is not supported by Object <XYZ> mean?”

This error message is thrown by the genIL layer if the user interfaces requests a relation that is unknown in genIL.

This could be caused by an invalid Wire Configuration between UIBBs of the Overview Page. You should first check if both the genIL model and the OVP wiring are consistent.

In productive customer systems it could be the case that the shared memory component used by genIL is outdated or corrupt (genIL uses shared memory only in productive systems but not in development or test system). You can use transaction SHMM to invalidate the shared memory area CL_CRM_GENIL_MODEL_SHM_AREA. Afterwards simply restart the user interface.

9.2 Why does the Undo button not work for key fields on the Overview Page?

All pre-defined MDG-F entities with SUT1 do not allow key changes. This rule applies to choosing the Undo button as well. After you enter a valid value for a key of the main entity, you cannot undo this action. You can also not undo any attribute change you have done at the same time as the key change.

The MDG-F 7.0 Feature Pack offers changing IDs. The how to guide Enable changeable IDs in MDG-F explains the new functionality in more details.

9.3 How can I search for an Edition instead of a Valid On date?

All database searches of the MDG-F user interfaces support both searching for specific validity dates or editions. The pre-defined user interface configurations use only the validity dates. If you want to switch to the edition instead, you need to change the application configuration of the related user interface.

Copy a pre-defined application configuration (for example MDGF_0G_OVP_CCTR for Cost Centers) to your custom namespace.

Edit the application parameter USMD_SEARCH_EDITION_MODE and set its value to X.

Note that you might have to adapt existing links (for example in the work centers) accordingly.

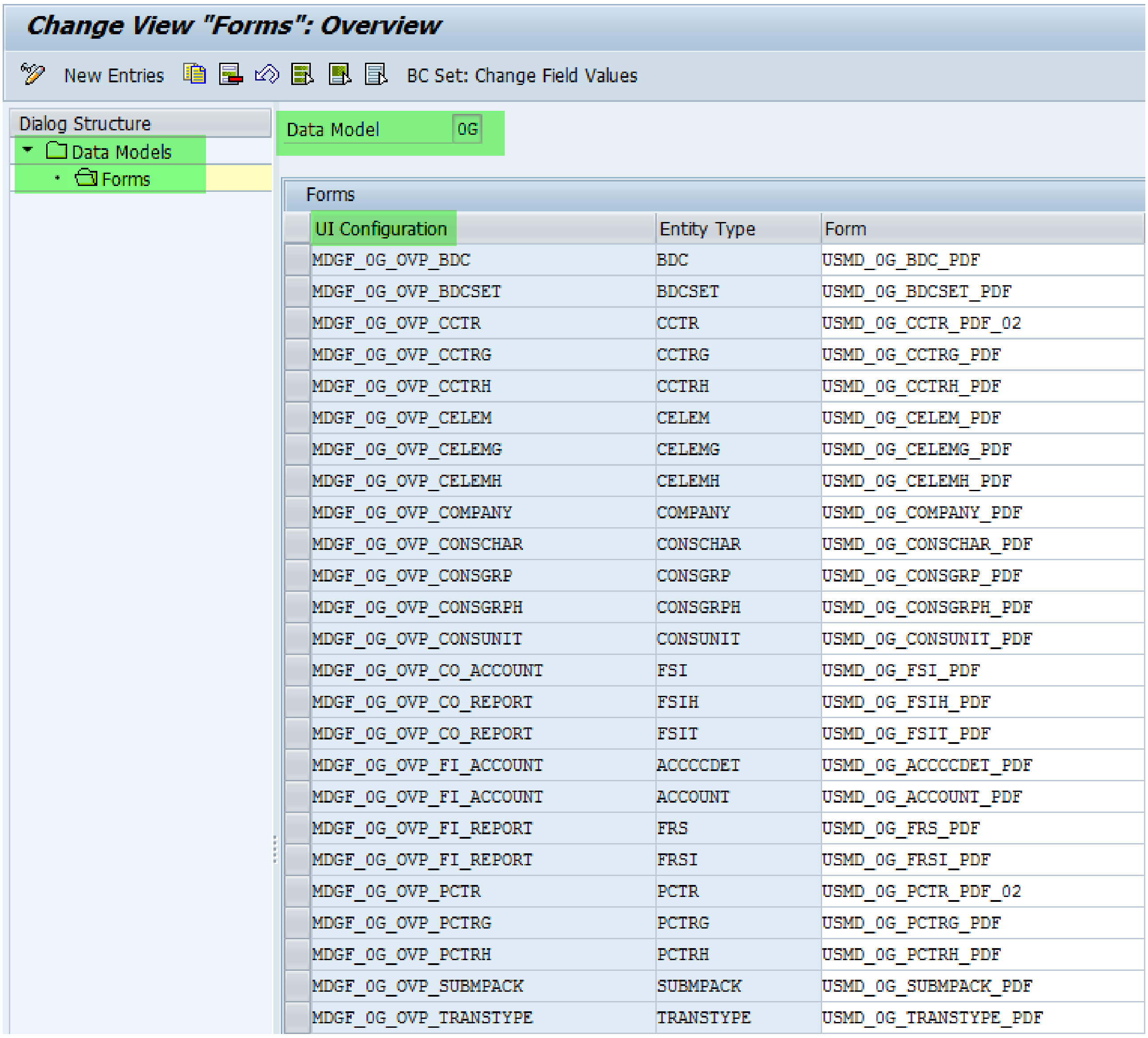
9.4 Why does the Search not find my Object that I have saved as Draft?

The minimal requirements for saving an object as a draft are the object keys. Once the keys are defined, saving is possible. Nevertheless, MDG does not create an inactive record in that case but stores the object only within the object list of a change request. The object list is not accessed by the search. The search can only display active and inactive records. You have to access a draft that contains only the object key via the My Change Request application. Alternatively ensure to maintain at least a single object attribute besides the object keys before saving the object as draft.

9.5 Why does the Print Preview not work in my copied Application Configuarion?

The print forms that are used for the print preview functionality are bound to the application configuration. If you have copied the application configuration to a custom version, you need to create new entries in the related customizing table:

- Start transaction MDGIMG and navigate to General Settings à UI Modeling à Assign Print Forms for Single Processing.
- Mark data model 0G and double click on Forms.
- Add entries using the name of your custom application configuration. The screenshot below shows the default values pre-defined by SAP.



Change View "Forms": Overview

Data Model 0G

Forms

UI Configuration	Entity Type	Form
MDGF_0G_OVP_BDC	BDC	USMD_0G_BDC_PDF
MDGF_0G_OVP_BDCSET	BDCSET	USMD_0G_BDCSET_PDF
MDGF_0G_OVP_CCTR	CCTR	USMD_0G_CCTR_PDF_02
MDGF_0G_OVP_CCTRG	CCTRG	USMD_0G_CCTRG_PDF
MDGF_0G_OVP_CCTRH	CCTRH	USMD_0G_CCTRH_PDF
MDGF_0G_OVP_CELEM	CELEM	USMD_0G_CELEM_PDF
MDGF_0G_OVP_CELEMG	CELEMG	USMD_0G_CELEMG_PDF
MDGF_0G_OVP_CELEMH	CELEMH	USMD_0G_CELEMH_PDF
MDGF_0G_OVP_COMPANY	COMPANY	USMD_0G_COMPANY_PDF
MDGF_0G_OVP_CONSCHAR	CONSCHAR	USMD_0G_CONSCHAR_PDF
MDGF_0G_OVP_CONSGRP	CONSGRP	USMD_0G_CONSGRP_PDF
MDGF_0G_OVP_CONSGRPH	CONSGRPH	USMD_0G_CONSGRPH_PDF
MDGF_0G_OVP_CONSUNIT	CONSUNIT	USMD_0G_CONSUNIT_PDF
MDGF_0G_OVP_CO_ACCOUNT	FSI	USMD_0G_FSI_PDF
MDGF_0G_OVP_CO_REPORT	FSIH	USMD_0G_FSIH_PDF
MDGF_0G_OVP_CO_REPORT	FSIT	USMD_0G_FSIT_PDF
MDGF_0G_OVP_FI_ACCOUNT	ACCCCDET	USMD_0G_ACCCCDET_PDF
MDGF_0G_OVP_FI_ACCOUNT	ACCOUNT	USMD_0G_ACCOUNT_PDF
MDGF_0G_OVP_FI_REPORT	FRS	USMD_0G_FRS_PDF
MDGF_0G_OVP_FI_REPORT	FRSI	USMD_0G_FRSI_PDF
MDGF_0G_OVP_PCTR	PCTR	USMD_0G_PCTR_PDF_02
MDGF_0G_OVP_PCTRG	PCTRG	USMD_0G_PCTRG_PDF
MDGF_0G_OVP_PCTRH	PCTRH	USMD_0G_PCTRH_PDF
MDGF_0G_OVP_SUBMPACK	SUBMPACK	USMD_0G_SUBMPACK_PDF
MDGF_0G_OVP_TRANSTYPE	TRANSTYPE	USMD_0G_TRANSTYPE_PDF

10 FAQ – Data Replication

In the following FAQs, ERP is a synonym for for SAP ERP and S/4 HANA systems.

10.1 How does the MDG Hub handle key mapping for MDG-F entities?

Key mapping is not written automatically, for example it is not written during data import or data replication. If you require key mapping in your replication scenarios, you have to define the key mapping for each object manually. The key mapping maintenance can be accessed via the common user interface using its Show button.

10.2 Why are attachments not replicated to ERP?

Most of the MDG-F entities support attachments. This enables adding documents and/or links to the master data. The attachments are a MDG specific functionality. They are not related to the SAP ECC document features and thus not part of data replication. Furthermore the IDOC and service interfaces of the financial objects do not support attachments.

10.3 Why don't ERP transactions show a change history for master data changed in MDG?

When object changes in MDG are replicated to ERP via IDOCs or services the ERP inbound works with a simple and efficient logic: all existing data are being deleted before the changed data coming from MDG are inserted. Unfortunately, no change documents are created for the deletion and the insert.

For this reason, we recommend using the change tracking in MDG when you run central master data maintenance.

10.4 How can I replicated MDG-F entities into the same client of my system?

MDG-F entities are completely stored within the generated tables of MDG. If you want to use the same system and client for productive usage of the data within SAP ERP, too, you need to replicate the MDG-F data into the same system and client. How-to guide "ALE Replication from MDG Hub to ERP using the Same Client in MDG-F" explains the required configuration.

10.5 Why are some of my Account IDocs not processed immediately?

The ALE IDoc based replication might show some IDocs with status 64 and the following messages in the target system:

- IDoc: <IDoc Number> Status: IDoc ready to be transferred to application
- Immediate Processing was Canceled

This might happen if the replication contained both the creation and change of accounts or accounts in company codes at the same time. If IDocs are processed in parallel by the receiving system, locking issues may occur.

We recommend using transaction WE02 in the target system to check the status of the IDocs. You can also use transaction BD87 to post-process the IDoc.

11 Additional Information

11.1 Further Reading

11.1.1 Information on SAP MDG on SAP S/4HANA

- Exchange knowledge: SAP Community | Q&A | Blog
- Try SAP Master Data Governance on S/4HANA for free: Trial Version
- Try SAP Master Data Governance on S/4HANA on the SAP Cloud Appliance Library: S/4HANA 2022 FPS1
- Learn more: Latest Release | Help Portal | How-to Information | Key Presentations

11.1.2 SAP Roadmap Explorer

- Please see the roadmap for SAP Master Data Governance

11.1.3 Related Information

- Learn more: Floorplan Manager for Web Dynpro ABAP | How to Adapt FPM | FPM Blog | How-to Information | Service Mapping Tool | SAP S/4HANA Cookbook CVI

11.2 SAP Notes

In addition to the detailed explanations written in this document, please see the following SAP Notes for further important information.

Note	Description
2021246	MDG-F: (Un-) Supported Fields in Data Model 0G
2337685	MDG-F: Standard Validations of Data Model 0G
3043582	MDG Customer Connection 2020
3194967	MDG Customer Connection 2021 for S/4HANA 2022
3311039	MDG Customer Connection 2023
1619534	How to Create, Enhance and Adapt FPM Applications
1637249	MDG: Information for efficient message processing
2105467	MDG Performance
2561461	Scope of support for SAP Master Data Governance (MDG)
1637249	MDG: Information for efficient message processing