SAP How-To Guide: Extend MDG-S Data Model by a new Entity Type (Flex Option)

Applies To:
SAP MDG-S / MDG-C running on SAP ECC 6 EhP 6 Master Data Governance. For more information, visit the Master Data Management homepage. (http://www.sdn.sap.com/irj/sdn/nw-mdm)

Summary
SAP Master Data Governance provides out-of-the-box solutions for the central management of master data objects. Domain-specific solutions include supplier governance (MDG-S), material governance (MDG-M), and financials governance (MDG-F).
If your domain-specific solution does not fully meet requirements, you can customize and extend it. You can use this guide to extend the MDG-S /MDG-C data model by a new entity type. The attributes of the new entity type only exist in the MDG context and not in the ERP data models (flex option).

Note: This guide describes an extension of the standard MDG Data Model, where data will be stored in MDG tables after activation. This is not the right guide for you if you need an extension where the data is stored in tables outside of MDG (for example Partner Functions).

Authors: Michael Theis, Lars Rueter
Company: SAP AG
Created on: March 2012
Version: 1.0
## Table of Contents

Introduction ............................................................................................................. 3

Scenario.................................................................................................................. 3

Technical Background ........................................................................................... 3

Implementation ....................................................................................................... 4
  Data Model Extension ............................................................................................ 5
  Create a new Entity Type ....................................................................................... 5
  Generate MDG Data Model-Specific Structures ................................................... 11

Extending the UI Configuration .............................................................................. 15
  Extend the genIL (Generic Interaction Layer) Model ........................................... 15
  Connect the MDG Data Model with the genIL Data Model .............................. 20

Copy standard MDG-UI configurations ................................................................. 22
  Copy Application Configuration for Supplier Overview page (OVP) ................ 22
  Copy Supplier Settings for MDG Communicator ................................................. 23
  Copy Supplier OVP ............................................................................................. 24

Create List UIBB .................................................................................................... 26
  Copy List UIBB ................................................................................................ 26
  Extend UIBB List Component ............................................................................. 28

Replace OVP Component in Application Configuration for Supplier .................. 32
  Add Custom List UIBB to Supplier OVP ............................................................ 33

Testing the Configuration ....................................................................................... 37

Related Content .................................................................................................... 39

Copyright ............................................................................................................... 40
**Introduction**

SAP Master Data Governance (MDG) is used for embedded MDM, that is, out-of-the-box, domain-specific master data governance to centrally create, change and distribute master data with a focus on SAP Business Suite.

Domain-specific content (data models, user interfaces, workflows) is provided as part of the standard for several application areas. Customers often want to adapt MDG data models to their needs.

This document explains how to extend the data model for supplier governance and the data model for customer governance by a new entity type. The attributes of the new entity type only exist in the MDG context and not in the ERP data models (flex option).

Before you work with this guide, we recommend you study the following how-to guide:

- Extensibility Options for SAP Master Data Governance -> Customer / Supplier Data -> How-to Guide: Extending the MDG Business Partner - Overview

**Scenario**

You want to extend the MDG data model for Business Partner by an additional entity type: **Purchasing Info Record**. You want this entity type to have a 1: N relationship with the Business Partner.

![Data Model - Supplier (Scope of 2011 Delivery) with custom entity type “Purchasing Info Record”](image)

**Technical Background**

This is a simple data model extension because the new field is only stored in MDG. There is no need to commit any data to ERP tables after activation of the change request. The activated data always remains in MDG and can only be used in this context.
**Implementation**

Two major building blocks make up the implementation of the entity type extension. In the first phase, you extend the MDG data model. In the second phase, you extend the User Interface to include the new entity type.

The flow diagram below shows the detailed implementation steps. Each box in the graphic corresponds to a section of the guide that includes detailed execution instructions.

*Figure: Implementation steps for flex extension*
**Data Model Extension**

You want to extend the MDG data model for Business Partner by an additional entity type *Purchasing Info Record*. You want the entity type to have a 1:N relationship with the Business Partner. The figure below shows how the data model looks in MDG.

You will first create a new Entity Type `ZINFOREC`. The Entity Type has two attributes `ZZPURBLOC` and `MATNR`. The relationship between `BP_HEADER` and `ZINFOREC` is 1:N of type Leading. There is a qualifying 1:N relationship between `ZINFOREC` and `MATNR`.

![Data Model Extension Diagram](image_url)

*Figure: Data Model details for extension*

**Create a new Entity Type**

1. Log into system for cross-client maintenance.

2. Start Customizing for Master Data Governance (transaction **MDGIMG**).

   Go to *General Settings -> Data Modeling -> Edit Data Model*.

   Select data model `BP`.

   Double click on entity types.

   Click pushbutton **New Entries**.
3. Create New Entity Type
   
   **Entity Type:** ZINFOREC
   
   **Storage/Use Type:** Changeable via Other Entity Type
   
   **Reuse Area:** MDG
   
   Save your settings.
4. Create another New Entity Type
   **Entity Type:** ZMATNR  
   **Storage/Use Type:** Not Changeable via MDG  
   **Data Element:** MATNR  
   Save your settings.

5. Navigate to the Relationships node.  
   Click pushbutton *New Entries* to create a new relationship.
6. **From-Entity Type**: 
- **BP_HEADER**
- **Relationship**: ZZBP2INFR
- **To-Entity Type**: ZINFOREC
- **Relat. Type**: Leading
- **Cardinality**: 1:N

Save your changes.

7. Click pushbutton **New Entries** to create a new relationship.

- **From-Entity Type**: ZMATNR
- **Relationship**: ZZMAT2INF
- **To-Entity Type**: ZINFOREC
- **Relat. Type**: Qualifying
- **Cardinality**: 1:N

Save your changes.

8. Select the new **Entity Type** ZINFOREC and double-click on the **Attributes** view.
9. Choose the **Edit** pushbutton. Choose the **New Entries** pushbutton. Create a new attribute with the following values:

**Attribute:** ZZPURBLOC  
**Data Element:** SPERM_X

10. Save your changes.

11. Activate your data model changes.
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>12.</strong></td>
<td>A messages popup displays (see screenshot.)</td>
</tr>
<tr>
<td><strong>13.</strong></td>
<td>Choose the <strong>Adjust Staging Area</strong> as shown to adjust existing change requests.</td>
</tr>
<tr>
<td><strong>14.</strong></td>
<td>In the following steps, you verify if the MDG staging structures were successfully generated. Choose the <strong>Visualize Data Model</strong> pushbutton.</td>
</tr>
<tr>
<td><strong>15.</strong></td>
<td>Choose the <strong>Active Version</strong> pushbutton.</td>
</tr>
<tr>
<td><strong>16.</strong></td>
<td>The screenshot shows the generated structures.</td>
</tr>
</tbody>
</table>
To view the generated tables, start transaction SE38.

Enter program USMD_DATA_MODEL.
Enter data model BP.
Run the program.

Double click on ZINFOREC.

Your table should look similar to the one shown in the screenshot.

Generate MDG Data Model-Specific Structures

After you change the MDG data model, you must regenerate the tables. In this customizing activity, for each data model and entity type, you generate technical structures and tables in the ABAP Dictionary. The system uses these structures internally for implementing the staging area. To generate these Data Model-specific structures follow the steps below.
In general, if you change a data model (for example, if you change attributes of entity types or relationships), you need to regenerate the structures.

1. Log into system for cross-client maintenance.

2. Start Customizing for Master Data Governance (transaction MDGIMG).

   Go to General Settings -> Data Modeling -> Generate Data Model-specific Structures
3. Create two **New Entries** as shown in the screenshot.
   
   One entry for **Active Area Mapping**.
   
   On entry for **Field Properties**.
   
   Save your changes.

4. Select the row with data model **BP**
   
   Double-click **Structures** in the left hand panel
   
   Click **Generate Structures**.

5. Verify that your structures for **ZINFOREC** were successfully generated.
6. In the following steps, you verify that the active area mapping structure was successfully generated.

Start transaction SE11.

Display structure ZXX_S_BP_FP_ZINFOREC by entering the details as shown.

7. You have now verified that the structure has been generated.
Extending the UI Configuration

To display the new fields in the Supplier-UI you first need to create a new Form-UIBB (User Interface Building Block) and add it to the UI component configuration BS_SP_OVP. You can use any of the following FPM options:

- Enhancement, adaptation, or customization of the existing component
- Adaptation of the component
- Copying of the UI configuration ZZBS_SP_OVP (as shown in this guide)

**Note**
For information on how to adapt the user interface using the floorplan manager; including advantages, disadvantages, and steps required, see SAP Note 1619534

Extend the genIL (Generic Interaction Layer) Model

1. Create an enhancement for the standard BUPA genIL Model (transaction code: genIL_model_browser)

   ![GenIL Model Editor](image)

   1. Create

2. Enter **Superenhancement**: BUPA_CUSP.

   ![Create Enhancement](image)
3. Verify the details of the enhancement.

4. Save the enhancement.

5. Choose the **Edit** pushbutton.
   In the *Model* tree, select **Dependent Object -> Create Dependent Object**.
6. Enter the name of the dependent object as shown.

![GenIL Model Editor: Change Component](image)

7. You must assign a **Key Structure** and an attribute structure to the dependent object.

   For the **Attribute Structure**, you enter the name of the active area mapping structure that you have generated in step 23 above (*ZXX_S_BP_PP_ZINFOREC*).

   For the Key Structure you must enter the name of a structure that you create in the next step (*ZINFOREC_KEY*).
8. Start transaction SE11.
Create structure ZINFOREC_KEY as shown.

- **Component**: INCLUDE
  - Reference Type: Types
  - Component Type: BSS_BPIL_ROOT_KEY
- **Component**: ZMATNR
  - Reference Type: Types
  - Component Type: MATNR

Save your changes.
Activate the structure.

9. Verify that the model nodes look similar to the ones shown.
10. Switch to edit mode. 
   Navigate to the `Relations` node. Right-click and select `Create Relations`.

11. Enter the details as shown.

---

### SAP How-To Guide: Extend MDG-S Data Model by a new Entity Type (Flex Option)

10. Switch to edit mode.
   Navigate to the **Relations** node. Right-click and select **Create Relations**.

11. Enter the details as shown.
12. In the relations detail screen select the assigned object **ZINFOREC**.

13. Save your changes.

---

**Connect the MDG Data Model with the genIL Data Model**

Create entries in the **VC_MDG_BS_GENIL_C** view cluster to relate the genIL model component to the entity type of the MDG data model.
1. Start transaction SE34. Display View Cluster VC_MDG_BS_GENIL_C

2. Choose the New Entries pushbutton.

3. Create a new entry with the following details:
   - Comp. Name: BUPA
   - Object Type: BUS1006
   - External Object: BP_Root

4. Mark the new entry and select Comp – GenIL - Node. Then choose the New Entries pushbutton.

5. Create a new entry with the following details:
   - Comp. Name: BUPA
   - External Object: ZINFOREC
   - Entity Type: ZINFOREC
Copy standard MDG-UI configurations

Copy Application Configuration for Supplier Overview page (OVP)

1. Start transaction SE80.
   Navigate to the Application Configuration as shown.
   Choose the Start Configurator pushbutton.

2. Copy the existing configuration to ZBS_OVP_SP.
Copy Supplier Settings for MDG Communicator

1. Start transaction SE80.
   Navigate to the Component Configuration for Web Dynpro application MDG_BS_GOV_COMMUNICATOR.
   Choose the Start Configurator pushbutton.

2. Copy the existing configuration to ZBS_OVP_SP.
   It is important that you use the same Name as the one in the application configuration in the previous section (ZBS_OVP_SP). This makes sure that you get the change request header in your Z application.
Copy Supplier OVP

1. Start transaction SE80.
   Navigate to the **Component Configuration** as shown.

2. Choose the **Start Configurator** pushbutton.
3. Copy the existing configuration to ZZBS_SP_OVP.

4. Refresh the list of configurations and look for your copy. It should be there.
## Create List UIBB

### Copy List UIBB

1. **Start transaction SE80.**
   
   Navigate to the *Component Configuration* as shown.

   ![Component Configuration](image1)

2. **Locate the FPM_LIST_UIBB_TEMPLATE in the list of component configurations.**
   
   Choose the *Start Configurator* pushbutton.

   ![Start Configurator](image2)
3. Copy the existing configuration to ZFPM_LIST_UIBB_PIR.

4. Refresh the list of configurations and look for your copy. It should be there.
Extend UIBB List Component

1. Start transaction SE80.
   Navigate to the *Component Configuration* as shown.

2. Locate your own `ZFPM_LIST_UIBB_PIR` configuration in the list of component configurations.
3. Choose the **Start Configurator** pushbutton.

4. Choose the **Continue in Change Mode** pushbutton.

5. Choose the **Feeder Class** pushbutton and enter the feeder class as shown.
6. Enter the feeder class parameters as shown.

7. Enter the general settings of the UIBB as shown.

8. Add your fields as table columns as shown.
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td><strong>Save your changes.</strong></td>
</tr>
<tr>
<td></td>
<td><img src="image1" alt="Component Configuration ZFPM_LIST_U1BB_PIR" /></td>
</tr>
<tr>
<td>10.</td>
<td><strong>Nothing needs to be added to the Toolbar Schema tab page.</strong></td>
</tr>
<tr>
<td></td>
<td><img src="image2" alt="Component Configuration ZFPM_LIST_U1BB_PIR" /></td>
</tr>
</tbody>
</table>
Replace OVP Component in Application Configuration for Supplier

1. Start transaction SE80.
   Navigate to your Application Configuration as shown.

2. Choose the Continue in Change Mode pushbutton.

3. Select the row starting with OVP and choose the pushbutton Assign Configuration Name.
4. Change the configuration name as shown.

![Configuration Name](image)

5. Save your changes.

Click on the link ZZBP_SP_OVP to edit the configuration.

![Configuration Name](image)

---

**Add Custom List UIBB to Supplier OVP**

1. Open the pushbutton **UIBB** menu, as shown.

![UIBB Menu](image)
2. Select **Freestyle Component** from the drop down as shown.

3. The empty property list appears.

4. Enter the attribute values as shown.
5. Choose the **Toolbar Schema** tab.

6. Choose the **Toolbar Element** pushbutton.
   Choose the **Button** pushbutton.
   Change the name of the pushbutton to **New** and assign the CREATE event to the pushbutton.
7. Enter the attribute values as shown.

8. You need to add an Event Parameter in the table. If you do not enter this event parameter you will later have problems in the UI with values disappearing after entry.

Click on the Add Parameter pushbutton and add the following parameter:

*Parameter Name:* DEFERRED_SENDING  
*Parameter Value:* X
9. Change to the Wire Schema tab. Choose the **Wire** pushbutton.

![Wire Schema tab](image1)

10. Enter the attribute values as shown.

   **Standard Attributes**
   - **Component**: FPM_LIST_UIBB
   - **Configuration Name**: ZFPM_LIST_UIBB_PIR
   - **Instance ID**: 
   - **Source Component**: FPM_FORM_UIBB_GL2
   - **Source Config Name**: BS_BP_ROOT
   - **Srce Inst. ID**: 
   - **Port Type**: Lead Selection
   - **Port Identifier**: STANDARD
   - **Connector Class**: *CL_BS_BP_CONNECTOR_BOL_REL*

   **Connector Parameters**
   - **Relation Name**: zbp2inforec
   - **Creation Mode**: Creation with Default Values

![Attributes of Wire](image2)

**Testing the Configuration**

To test your configuration start the MDG Supplier UI using the following URL and replace the parameter value `WDCONFIGURATIONID` with the name of your copy of the standard configuration.

**URL:**

http://<host>:<port>/sap/bc/webdynpro/sap/bs_ovp_bp?sap-client=405&sap-language=EN&WDCONFIGURATIONID=ZBS_OVP_SP
**Organization: $118, (no description available)**

- **ERP Vendors**
  - Replication Status:
    - Actions: *ERP Vendor*
    - Reason: Standard
    - The table does not contain any data
  - New

- **Roles**

- **Addresses**

- **Address Usages**

- **Bank Accounts**
  - Replication Status:
    - Actions: *ID, Bank Key, Bank Name, Bank Account, Account Name, Valid*
    - The table does not contain any data

- **Identification Numbers**

- **Tax Numbers**

- **Industries**

- **Purchasing Information Record**
  - Replication Status:
    - Actions: *Material*
    - Control purchasing block
    - The table does not contain any data

---

*SAP COMMUNITY NETWORK*  
© 2012 SAP AG  
*scn.sap.com*  
38
Related Content

1. For more information, visit the Master Data Management homepage. (http://www.sdn.sap.com/irj/sdn/nw-mdm)
Copyright

© Copyright 2012 SAP AG. 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 AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2 Universal Database, System i, System i5, System p, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group 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.