**Apply 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](http://www.sdn.sap.com/irj/sdn/nw-mdm))

**Summary**
SAP Master Data Governance provides an out-of-the-box solution for the central management of various master data objects such as financial objects, supplier, and material. But SAP Master Data Governance also provides the flexibility to customize the solution, when pre-delivered content does not fully match customer requirements. You can use this guide to extend the MDG-S data model or the MDG-C data model by adding attributes that already exist as database fields in the appropriate customer include of the SAP Business Partner / Vendor / Customer (MDG reuse option).

**Author:** Lars Rueter

**Company:** SAP AG

**Created on:** May 2014

**Version:** 1.2
Table of Contents

Introduction ........................................................................................................................................... 3
Prerequisites .......................................................................................................................................... 3
Scenario .............................................................................................................................................. 3
  High Level Requirements .................................................................................................................. 4
  Governance Process .......................................................................................................................... 4
Data Model ........................................................................................................................................... 4
  Test Your Configuration .................................................................................................................... 5
Implementation ................................................................................................................................. 8
  Extended Field Exists in Business Partner Only ......................................................................... 9
    Extend the BUT00 Data Model .................................................................................................... 10
    Extend the MDG Data Model ....................................................................................................... 11
    Generate Data Model-Specific Structures for MDG ................................................................. 15
    Define SMT Mapping ................................................................................................................... 18
    Adjust Staging Area of Linked Change Requests .................................................................... 26
    Test Your Configuration .............................................................................................................. 27
  Extended Field Exists in Vendor / Customer Only .................................................................... 29
    Extend the MDG Data Model ....................................................................................................... 30
    Generate Data Model-Specific Structures for MDG ................................................................. 33
    Define SMT Mapping ................................................................................................................... 35
    Test Your Configuration .............................................................................................................. 40
  Extended Field Exists in Business Partner and in Vendor / Customer .................................... 42
    Extend LFA1 .................................................................................................................................. 44
    Implement Enhancement Spot CVI_CUSTOM_Mapper ............................................................... 46
    Test Your Configuration .............................................................................................................. 48
User Interface Configuration ............................................................................................................ 50
  Adaptation Options in Floorplan Manager .................................................................................. 51
  Adding a New Field to the Supplier UI for Single Maintenance (BUT00 Extension Case) ........... 52
  Adding a New Field to the Supplier UI for Single Maintenance (Extended Field Exists in Vendor / Customer Only) ......................................................................................................................... 54
Related Content ............................................................................................................................ 57
Copyright ................................................................................................................................ .......... 58
Introduction

SAP Master Data Governance (MDG) is used for embedded Master Data Management (MDM), that is, centralized, out-of-the-box, domain-specific creation, modification, and distribution of 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. It is a common requirement from customers to adapt the MDG data models to their specific needs.

This guide explains how to extend the MDG-S/C data model by adding a single field to an existing entity.

Prerequisites

You must have access to a sandbox of your MDG Hub Application Server with appropriate permissions.

You must have access to the MDG-Hub client and to the client in which you do the cross system customizing.

The ERP System must run on Enhancement Package 6 (EhP6).

You must have basic understanding of the following areas:

- SAP ERP
- SAP MDG
- SAP Floorplan Manager (FPM)

Scenario

You want to extend the MDG data model for Business Partner by an additional field “License”. The field should be added to the Central Data entity type as shown in the figure below.

Figure: Data Model – Supplier (Scope of 2011 Delivery)
High Level Requirements
The business requires the new field “License” as part of the MDG Business Partner data model.

Governance Process
The default governance process delivered with MDG is used. No changes to the governance process are necessary as part of this scenario.

Comprehensive Data Maintenance Requires Involvement of Many People

Data Model
In SAP Master Data Governance data can be stored in the staging area or in the active area. The staging area temporarily stores the working version of master data that is undergoing change within a change request process. The active area permanently stores active instances of master data after changes have been approved by a change request process.

Extending the data model in SAP Master Data Governance with custom attributes or tables involves changes to the data structures in the active area and in the staging area. It also involves defining a mapping between these data structures.
You can extend the MDG-S / MDG-C data model by adding attributes that already exist as database fields in the appropriate ERP tables. This method of extending the data model is called the MDG reuse option.

Your implementation of the reuse option depends on the ERP tables that store the master data after it is activated. The following options are available:

- **Extended Field Exists in Business Partner**
  The new field is only extended as a database field in the following customer includes of the SAP Business Partner: CI_EEW_BUT000 and CI_EEW_BUT000_X

- **Extended Field Exists in Vendor / Customer**
  The new field is only extended as a database field in the customer include *Ext. Interface: Central Vendor Data / Data Fields* (VMDS_EI_VMD_CENTRAL_DATA) for the vendor data model as well as in either the customer include *Ext. Interface: Central Vendor Data / X-Fields* (VMDS_EI_VMD_CENTRAL_DATA_XFLAG) for the vendor data model or in the equivalent customer include for the customer data model.

**Test Your Configuration**

To test your configuration you should first add your new field to the MDG single maintenance UI. Section *User Interface Configuration* provides some instructions on how to add the new field to the UI.

Create a new change request in MDG and enter a value for the new field in the ERP Vendor Control Data section. Then process the change request until it is activated. Write down the ID of the new Supplier after activation.

Run transaction **SE16** and find your newly created Vendor in table LFA. Make sure the field **ZZ_LICENSE** is there and contains the right value.
### Table LFA1 Display

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSOVN</td>
<td></td>
</tr>
<tr>
<td>PSOTL</td>
<td></td>
</tr>
<tr>
<td>PSOHS</td>
<td></td>
</tr>
<tr>
<td>PSOST</td>
<td></td>
</tr>
<tr>
<td>TRANSPORT CHAIN</td>
<td></td>
</tr>
<tr>
<td>STAGING TIME</td>
<td>0</td>
</tr>
<tr>
<td>SCHEDULING TYPE</td>
<td></td>
</tr>
<tr>
<td>SUBMI RELEVANT</td>
<td></td>
</tr>
<tr>
<td>ZMASCHNI</td>
<td></td>
</tr>
<tr>
<td>ZSDN CI</td>
<td></td>
</tr>
<tr>
<td>ZZ SECRET ALIAS</td>
<td></td>
</tr>
<tr>
<td>ZZ LICENSE</td>
<td>ND-YY 3333</td>
</tr>
<tr>
<td>ZZLOCVEN</td>
<td>ND-XX 2222</td>
</tr>
</tbody>
</table>

Filled in by CVI

Filled by SMT mapping from MDG
Extended Field Exists in Business Partner and in Vendor / Customer

The new field is extended as a database field in both the Business Partner and in customer includes for vendor / customer.

Each of the sections in this document contains detailed step-by-step implementation instructions and you should follow the steps in the section that is appropriate for your scenario.

These cases are also shown in the figure below.

![Figure: Different cases for extending the MDG BP data model](image-url)

**Note**

You can create new attributes in the customer namespace, for example, under ZZ* and YY*.
Implementation

Even though we have to distinguish three cases for the field extension, the implementation steps are very similar. The implementation involves the following main tasks:

- Extending the ERP data model
- Extending the MDG data model
- Generating the MDG data model specific structures
- Defining SMT mapping
- Implement CVI mapping (if applicable)

The main differences in the implementation approaches can be identified from the table below. During the step “define SMT mapping” it is important to map to the right structures. If the field extension was made in both BP and vendor / customer you must additionally implement the CVI mapping (but only SMT mapping to BP is necessary).

<table>
<thead>
<tr>
<th>Implementation Steps</th>
<th>Field only in Business Partner (BP)</th>
<th>Field only in Vendor / Customer</th>
<th>Field in both BP and Vendor / Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extend ERP data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Extend MDG data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Generate MDG data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Define SMT mapping</td>
<td>✓ ¹</td>
<td>✓ ²</td>
<td>✓ ¹</td>
</tr>
<tr>
<td>Implement CVI mapping</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

*Table: Implementation steps for extending the MDG-S/C data model by a new field*

<table>
<thead>
<tr>
<th>SMT Mappings</th>
</tr>
</thead>
<tbody>
<tr>
<td>¹ MDG staging ← Business Partner</td>
</tr>
<tr>
<td>² MDG staging ← Vendor / Customer</td>
</tr>
</tbody>
</table>

*Table: SMT mappings for different extension cases*
Extended Field Exists in Business Partner Only

In this section, you learn how to add a new field “License” to the MDG BP data model. In the figure below you can see that it is necessary to extend the MDG BP data model. You also must have extended the BUT000 table on the MDG hub and client systems. In order to facilitate data transfer between the MDG staging area and the BUT000 table on the MDG hub you have to extend the SMT mapping.

![Diagram showing data flow between systems](image)

This section is subdivided in to one sub-section for each implementation step. Take a look at the table below to get an overview of the steps involved.

<table>
<thead>
<tr>
<th>Implementation Steps</th>
<th>Field only in Business Partner (BP)</th>
<th>Field only in Vendor / Customer</th>
<th>Field in both BP and Vendor / Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extend ERP data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Extend MDG data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Generate MDG data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Define SMT mapping</td>
<td>✓¹</td>
<td>✓²</td>
<td>✓¹</td>
</tr>
<tr>
<td>Implement CVI mapping</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Table: Implementation steps for extending the MDG-S/C data model by a new field
When you get to the implementation step where SMT mapping is required you should refer to the table below to find the mappings you need to extend. The SMT package group can help you identifying the right mapping.

<table>
<thead>
<tr>
<th>SMT Mappings</th>
<th>SMT Package Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 MDG staging ↔ Business Partner</td>
<td>MDG_BS_BP</td>
</tr>
<tr>
<td>2 MDG staging ↔ Vendor / Customer</td>
<td>MDG_BS_ECC_CUST</td>
</tr>
<tr>
<td></td>
<td>MDG_BS_ECC_SUPPL</td>
</tr>
</tbody>
</table>

Table: Details regarding SMT mappings for different extension cases

Extend the **BUT000** Data Model

The procedure below describes how to extend the append structures for supplier (**BUT000**). This involves adding the **ZZ_LICENSE** field to the **CI_EEW_BUT000** structure and the **CI_EEW_BUT000_X** structure.

1. Log into system for cross-client maintenance.
2. Start transaction **SE11**.
   - Open structure **CI_EEW_BUT000**.
   - Add the new field **ZZ_LICENSE** to the structure.
   - Activate the changes. (Be patient. The activation might take a while).
   Open structure BUT000.
   Verify that the new field "ZZ_LICENSE" is included in the BUT000 structure.

   Open structure CI_EEW_BUT000_X.
   Add the new field ZZ_LICENSE to the structure.
   Activate the changes. (Be patient. The activation might take a while).

---

**Extend the MDG Data Model**

5. Log into system for cross-client maintenances.
6. Start Customizing for **Master Data Governance** (transaction MDGIMG).
   Select **General Settings -> Data Modeling -> Edit Data Model.**

7. Select the **BP** data model.

8. Double-click **Entity Types.**
   Select **Entity Type** BPBP_CENTRL.
9. Double-click **Attributes**.

Choose the **New Entries** pushbutton to add a new attribute to the **BPBP_CENTRL** entity type.

10. Enter the field name **ZZLICENSE** and an appropriate data element (for example, **ZZ_LICENSE**).
11. **Activate the changes.**

![Image showing Change View "Attributes": Overview]

12. **Start transaction SE38.**

Run program USMD_DATA_MODEL.

![Image showing ABAP Editor: Initial Screen]
13. Locate entity type BP_CENTRAL and double-click it to see the generated table.

14. Verify that the generated table contains the added field /1MD/BPZZLICENSE.

---

**Generate Data Model-Specific Structures for MDG**

Whenever you change the MDG data model, you need to regenerate the tables. In this customizing activity, you generate technical structures and tables in the ABAP Dictionary for each data model and entity type. The
system uses these structures internally for implementing the staging area. To generate these data model-specific structures follow the steps below.

**Note**
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.

15. Log into the system for cross-client maintenance.


Select **General Settings -> Data Modeling -> Generate Data Model-Specific Structures.**
17. Select the row with **BP Data Model**.
   Double-click **Structures** in the left hand panel.
   Choose the **Generate Structures** pushbutton.

18. In the following steps, you verify that your field extension was successfully added to the **Customer Include** of the mapping structure.
   Start transaction SE11.
   Display structure `/MDGBP/_S_BP_PP_BPBP_CENTRL` by entering the details as shown.
19. The new field has been generated in the Customer Include of the mapping structure.

### Define SMT Mapping

The Service Mapping Tool (SMT) is a program that enables you to fill target structures by using sets of source structures (in ABAP). It supports simple mappings, mappings with field transformations, complex transformations, and field checks.

You need to perform the following mappings:

- **Mapping Central Data to the API (BP_BBP_CENTRL_2API)**
  - Central Business Partner Data (starting from step 20 below)
  - X Structure for Central Business Partner Data (starting from step 25 below)

- **Mapping Central Data to Staging (BP_BBP_CENTRL_2STA)**
  - Central Business Partner Data (starting from step 31 below)

20. Log into system for cross-client maintenances.

Select General Settings -> Data Modeling -> Extend Mappings -> Extend Mappings.

Select Mapping -> Open.

Enter BP_BP_CENTRAL_2API.

22. You have opened the mapping of BP central data from MDG staging area to the service API.

You see a table with mapping steps.

You start by extending the mapping for the Central Business Partner Data.
23. Select the first mapping step as shown. Choose the Details pushbutton.

24. Select the Transformations tab.
25. Change to edit mode by choosing the **Edit** pushbutton.

   Above the transformations table, choose the **Add** pushbutton to add a new transformation.

26. Choose the **Add** pushbutton above the **Field Mapping** table.

   Enter the mapping for **ZZ_LICENSE** as shown.

   Save the changes.
27. Return to the Mapping Step table for the mapping BP_BP_CENTRAL_2API.

You have opened the mapping of BP central data from MDG staging area to the service API.

You see a table with mapping steps.

You will now extend the mapping for the Central Business Partner Data X structure.

Select the `_XFLAG` mapping step and choose the Details pushbutton.

28. Select the Transformations tab.
29. Change to edit mode by clicking the **Edit** pushbutton.
   Above the transformations table choose the **Add** pushbutton to add a new transformation.

<table>
<thead>
<tr>
<th>Field Check</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>Transformation Type</td>
</tr>
<tr>
<td>00001</td>
<td>Field Mapping</td>
</tr>
<tr>
<td>00001</td>
<td>Field Mapping</td>
</tr>
</tbody>
</table>

30. Choose the **Add** pushbutton above the **Field Mapping** table.
31. Enter the mapping for **ZZ_LICENSE** as shown.

32. Save the changes.

33. Select **Mapping -> Open**.
Enter **BP_BP_CENTRAL_2STA**.
34. You have opened the mapping of BP central data from the service API to the MDG staging area.

You see a table with mapping steps.

You need to extend the mapping for the Central Business Partner Data.

35. Select the **Transformations** tab.

Choose the **Edit** pushbutton.

Choose the **Add** pushbutton to add a new transformation.
36. Choose the **Add** pushbutton above the **Field Mapping** table.

Enter the mapping for **ZZLICENSE** as shown.

Save your changes.

---

**Adjust Staging Area of Linked Change Requests**

During the activation of data models, you can prevent data loss due to the addition of fields from the active area. To do this, you must adjust the relevant reuse access class after the data model has been changed and activated so that the field that has been added can be read from the active area. In each client of the relevant systems that contains the change request to be adjusted, you must perform the following steps.

37. Log into system for which change request adjustment is required.

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

Select **General Settings -> Data Modeling -> Edit Data Model**.

Select data model BP and press the button "Adjust Staging Area of Linked Change Requests" as shown in the screenshot.
Test Your Configuration

To test your configuration you should first add your new field to the MDG single maintenance UI. Section *User Interface Configuration* explains how to add the new field to the UI.

Create a new change request in MDG and enter a value for the new field. Then process the change request until it is activated. Write down the ID of the new Supplier after activation.

Run transaction SE16 and find your newly created Business Partner in table BUT000. Make sure the field **ZZ_LICENSE** is there and contains the right value.
## Table LFA1 Display

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>J SC CAPITAL</td>
<td>00</td>
</tr>
<tr>
<td>J SC CURRENCY</td>
<td></td>
</tr>
<tr>
<td>AIC</td>
<td></td>
</tr>
<tr>
<td>PMT OFFICE</td>
<td></td>
</tr>
<tr>
<td>PSON1</td>
<td></td>
</tr>
<tr>
<td>PSON2</td>
<td></td>
</tr>
<tr>
<td>PSON3</td>
<td></td>
</tr>
<tr>
<td>PSON4</td>
<td></td>
</tr>
<tr>
<td>PSOVN</td>
<td></td>
</tr>
<tr>
<td>PSOTL</td>
<td></td>
</tr>
<tr>
<td>PSOHS</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td></td>
</tr>
<tr>
<td>TRANSPORT CHAIN</td>
<td></td>
</tr>
<tr>
<td>STAGING TIME</td>
<td>0</td>
</tr>
<tr>
<td>SCHEDULING TYPE</td>
<td></td>
</tr>
<tr>
<td>SUBMRELEVANT</td>
<td></td>
</tr>
<tr>
<td>ZZ LICENSE</td>
<td>XX</td>
</tr>
</tbody>
</table>
Extended Field Exists in Vendor / Customer Only

In this section you will learn how to add a new field *License* to the MDG BP data model. In the figure below you can see that it is necessary to extend the MDG BP data model. You also must have extended the Vendor / Customer table on the MDG hub and client systems. In order to facilitate data transfer between the MDG staging area and the LFA1 / KNA1 table on the MDG hub you have to extend the SMT mapping.

![Diagram showing data model and field extension](image)

**Figure: Field is added only to Vendor / Customer**

This section is subdivided into one sub-section for each implementation step. Take a look at the table below to get an overview of the steps involved.

<table>
<thead>
<tr>
<th>Implementation Steps</th>
<th>Field only in Business Partner (BP)</th>
<th>Field only in Vendor / Customer</th>
<th>Field in both BP and Vendor / Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extend ERP data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Extend MDG data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Generate MDG data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Define SMT mapping</td>
<td>✓¹</td>
<td>✓²</td>
<td>✓¹</td>
</tr>
<tr>
<td>Implement CVI mapping</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

*Table: Implementation steps for extending the MDG-S/C data model by a new field*

When you get to the implementation step where SMT mapping is required, refer to the table below to find the mappings you need to extend. The SMT package group can help you identify the right mapping.
### SMT Mappings

<table>
<thead>
<tr>
<th>SMT Mappings</th>
<th>SMT Package Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDG staging ↔ Business Partner</td>
<td>MDG_BS_BP</td>
</tr>
</tbody>
</table>
| MDG staging ↔ Vendor / Customer | MDG_BS_ECC_CUST
|                             | MDG_BS_ECC_SUPPL |

### Table: Details regarding SMT mappings for different extension cases

Since this scenario contains several similarities to the one explained in section *Extended Field Exists in Business Partner* and the one explained in section *Extended Field Exists in Vendor / Customer* on the detailed instructions below focus only on the additional steps.

<table>
<thead>
<tr>
<th>Extend ERP data model</th>
<th>For the extension of <strong>LFA1</strong> follow the steps in section <strong>Extend</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extend MDG data model</td>
<td>To extend the MDG data model, follow the steps below.</td>
</tr>
<tr>
<td>Generate MDG data model specific structures</td>
<td>To generate the MDG data model, follow the steps below.</td>
</tr>
<tr>
<td>Adjust Staging Area of Linked Change Requests</td>
<td>Follow steps 37-39 in the section <em>Adjust Staging Area of Linked Change Requests</em>.</td>
</tr>
<tr>
<td>Define SMT mapping</td>
<td>To define SMT mapping between MDG staging and vendor, follow the steps below.</td>
</tr>
</tbody>
</table>

### Extend the MDG Data Model

40. Log into system for cross-client maintenances.

41. Start Customizing for **Master Data Governance** (transaction **MDGIMG**). Select **General Settings -> Data Modeling -> Edit Data Model**.

![Screenshot of MDGIMG transaction](image)
42. Select data model **BP**.

43. Double-click entity types.
   Select entity **BP_VENGEN**.

44. Double-click Attributes.
    Choose **New Entries** to add a new attribute to the entity **BP_VENGEN**.
45. Enter the field name ZZLICVEN and an appropriate data element (for example, ZZ_LICENSE).

![Image showing New Entries: Overview of Added Entries]

46. Activate the changes.

![Image showing Change View "Attributes": Overview]

47. Start transaction SE38. Run program USMD_DATA_MODEL.

![Image showing ABAP Editor: Initial Screen]
48. Locate entity type BP_VENGEN and Double-click it to see the generated table.

49. Verify that the generated table contains the added field /1MD/BPZZLICVEN.

Generate Data Model-Specific Structures for MDG

Whenever you change the MDG data model, you need to regenerate the tables. In this customizing activity, you generate technical structures and tables in the ABAP Dictionary for each data model and entity type. The system uses these structures internally for implementing the staging area. To generate these data model-specific structures follow the steps below.

Note
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.
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.</td>
<td>Log into system for cross-client maintenances.</td>
</tr>
<tr>
<td>51.</td>
<td>Start Customizing for <strong>Master Data Governance</strong> (transaction <strong>MDGIMG</strong>). Select <strong>General Settings -&gt; Data Modeling -&gt; Generate Data Model-Specific Structures</strong>.</td>
</tr>
<tr>
<td>52.</td>
<td>Select the row with Data Model <strong>BP</strong>. Double-click <strong>Structures</strong> in the left hand panel. Choose the <strong>Generate Structures</strong> pushbutton.</td>
</tr>
</tbody>
</table>
In the following steps you will verify that your field extension was successfully added to the Customer Include of the mapping structure.

Start transaction SE11.

Display structure /MDGBPX/_S_BP_PP_BP_VENGEN by entering the details as shown.

The new field has been generated in the Customer Include of the mapping structure.

Define SMT Mapping

The Service Mapping Tool (SMT) is a program that enables you to fill target structures by using sets of source structures (in ABAP). It supports simple mappings, mappings with field transformations, complex transformations, and field checks.

You need to perform the following mappings:

- Mapping Central Data -> API (SUPPL_BP_VENGEN_2API)
  - Central Business Partner Data
  - X Structure for Central Business Partner Data
- Mapping Central Data -> Staging (SUPPL_BP_VENGEN_2STA)
  - Central Business Partner Data

55. Log into system for cross-client maintenances.

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

Select **General Settings -> Data Modeling -> Extend Mappings -> Extend Mappings**.

Select **Mapping -> Open**.

Enter **SUPPL_BP_VENGEN_2API**.
57. Select mapping structure VMDS_EI_VMD_CENTRAL_DATA.
Choose the Details pushbutton above the Mapping Steps table.

58. Select the Transformations tab.
Change to edit mode by clicking the Edit pushbutton.
Above the transformations table choose the Add pushbutton to add a new transformation of type Field Mapping.
59. Choose the **Save** pushbutton.

60. Select Mapping **SUPPL_BP_VENGEN_2API**.

61. Select the mapping step **Update flag for vendor central data**.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>62.</td>
<td>Choose the <strong>Save</strong> pushbutton.</td>
</tr>
</tbody>
</table>
| 63.  | Start Customizing for **Master Data Governance** (transaction MDGIMG).  
Select **General Settings** -> **Data Modeling** -> **Extend Mappings** -> **Extend Mappings**.  
Select **Mapping** -> **Open**.  
Enter **SUPPL_BP_VENGEN_2STA**. |
64. Select mapping structure MDG_BS_SUPPL_BP_VENGEN. Choose Details above the Mapping Steps table.

65. Select the Transformations tab. Change to edit mode by clicking the Edit pushbutton. Above the transformations table choose the Add pushbutton to add a new transformation of type Field Mapping.

Test Your Configuration

To test your configuration you should first add your new field to the MDG single maintenance UI. Section User Interface Configuration provides some instructions on how to add the new field to the UI.

Create a new change request in MDG and enter a value for the new field in the ERP Vendor Control Data section. Then process the change request until it is activated. Write down the ID of the new Supplier after activation.
Run transaction **SE16** and find your newly created Vendor in table LFA. Make sure the field **ZZ_LICENSE** is there and contains the right value.
Extended Field Exists in Business Partner and in Vendor / Customer

In this section you will learn how to add a new field “License” to the MDG BP data model. In the figure below you can see that it is necessary to extend the MDG BP data model. You also must have extended the BUT000 and LFA1 / KNA1 tables on the MDG hub and client systems. In order to facilitate data transfer between the MDG staging area and the BUT000 on the MDG hub you have to extend the SMT mapping. To facilitate data transfer from BUT000 on the hub and the LFA1 / KNA1 tables on the MDG hub you have to implement CVI mapping.

Figure: Field is added to Business Partner and Vendor / Customer

In this scenario the Customer Vendor Integration (CVI) is taking care of the data transfer between BUT000 and LFA1 / KNA1 data structures. For a better understanding you can take a look at the CVI architecture overview shown in the figure below.

Figure: Architecture Overview Customer Vendor Integration (CVI)
This section is subdivided in to one sub-section for each implementation step. Take a look at the table below to get an overview of the steps involved.

<table>
<thead>
<tr>
<th>Implementation Steps</th>
<th>Field only in Business Partner (BP)</th>
<th>Field only in Vendor / Customer</th>
<th>Field in both BP and Vendor / Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extend ERP data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Extend MDG data model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Generate MDG data model specific structures</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Define SMT mapping</td>
<td>✓ 1</td>
<td>✓ 2</td>
<td>✓ 1</td>
</tr>
<tr>
<td>Implement CVI mapping</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

*Table: Implementation steps for extending the MDG-S/C data model by a new field*

When you get to the implementation step where SMT mapping is required you should refer to the table below to find the mappings you need to extend. The SMT package group can help you identifying the right mapping.

<table>
<thead>
<tr>
<th>SMT Mappings</th>
<th>SMT Package Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 MDG staging → Business Partner</td>
<td>MDG_BS_BP</td>
</tr>
<tr>
<td>2 MDG staging → Vendor / Customer</td>
<td>MDG_BS_ECC_CUST MDG_BS_ECC_SUPPL</td>
</tr>
</tbody>
</table>

*Table: Details regarding SMT mappings for different extension cases*

Since this scenario is very similar to the one explained in section *Extended Field Exists in Business Partner* the detailed instructions below focus only on the additional steps.

<table>
<thead>
<tr>
<th>Extend ERP data model</th>
<th>For the extension of BUT000 follow steps 1-4 in section <em>Extend the BUT000 Data</em> Model starting. To extend LFA1 follow the steps outlined in the next section.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extend MDG data model</td>
<td>Follow the steps 5-14 in section <em>Extend the MDG Data</em> Model.</td>
</tr>
<tr>
<td>Generate MDG data model specific structures</td>
<td>Follow the steps 15-19 in section <em>Generate Data Model</em>.</td>
</tr>
<tr>
<td>Define SMT mapping</td>
<td>Follow steps 20-36 in the section <em>Define SMT Mapping</em>.</td>
</tr>
<tr>
<td>Adjust Staging Area of Linked Change Requests</td>
<td>Follow steps 37-39 in the section <em>Adjust Staging Area of Linked Change Requests</em>.</td>
</tr>
<tr>
<td>Implement CVI mapping</td>
<td>For implementing the CVI mapping follow the steps below.</td>
</tr>
</tbody>
</table>
## Extend **LFA1**

In this section you extend the append structures for vendor (LFA1). You add the field **ZZ_LICENSE** to structures **LFA1**, **VMDS_EI_VMD_CENTRAL_DATA** and **VMDS_EI_VMD_CENTRAL_DATA_XFLAG**.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.</td>
<td>Log into system for cross-client maintenances.</td>
</tr>
</tbody>
</table>
| 67.  | Start transaction **SE11**.  
Open structure **ZMDG_SUPPLIER_EHP6_TEST_SUP1**.  
Add the new field **ZZ_LICENSE** to the structure.  
Activate the changes. |
| 68.  | Verify that the appended field appears in structure **VMDS_EI_VMD_CENTRAL_DATA**. |
Start transaction SE11.
Open structure ZMDG_SUPPLIER_EHP6_TEST_SUPL2.
Add the new field ZZ_LICENSE to the structure.
Activate the changes.

Verify that the appended field appears in structure VMDS_EI_VMDentral_DATA_XFLAG.

Start transaction SE11.
Open structure LFA1.
Create an append structure and add the new field ZZ_LICENSE to the append structure.
Activate the changes.
### 72. Verify that the appended field appears in structure **LFA1**.

#### Dictionary: Display Table

- **Transp. Table**: LFA1
- **Short Description**: Vendor Master (General Section)

![Dictionary: Display Table](image)

### 73. Log into system for cross-client maintenances.

### 74. Start transaction **SE18**.

Enter **CVI_CUSTOM_MAPPER** and choose the **Display** pushbutton.

#### BAdI Builder: Initial Screen for Definitions

![BAdI Builder: Initial Screen for Definitions](image)

**Implement Enhancement Spot **CVI_CUSTOM_MAPPER**

To map our custom field from business partner to vendor and back you must implement interface **IF_EX_CVI_CUSTOM_MAPPER** of enhancement spot **CVI_CUSTOM_MAPPER**. You should at least implement the methods highlighted in the figure below.
75. Double-click **Implementations**.

76. Create a new BAdI implementation **ZZ_EXTENSIBILITY_TEST**.
77. You will need to implement the interface methods `MAP_BP_TO_VENDOR` and `MAP_VENDOR_TO_BP`.

78. `MAP_BP_TO_VENDOR`

\[\text{c\_vendor-central\_data-central\_data-\text{zz\_license}} = \text{i\_partner-central\_data-common\_data-\text{zz\_license}}.\]

\[\text{c\_vendor-central\_data-central\_data-\text{zz\_license}} = \text{i\_partner-central\_data-common\_data-\text{zz\_license}}.\]

79. `MAP_VENDOR_TO_BP`

\[\text{c\_partner-central\_data-common\_data-\text{zz\_license}} = \text{i\_vendor-central\_data-central\_data-\text{zz\_license}}.\]

\[\text{c\_partner-central\_data-common\_data-\text{zz\_license}} = \text{i\_vendor-central\_data-central\_data-\text{zz\_license}}.\]

**Test Your Configuration**

To test your configuration you should first add your new field to the MDG single maintenance UI. At the end of this document you will find a section that provides some instructions on how to add the new field to the UI.

Create a new change request in MDG and enter a value for the new field. Then process the change request until it is activated. Write down the ID of the new Supplier after activation. Check `BUT000` and `LFA1` the value for the new License field should appear in both tables.
User Interface Configuration

The UI is configured using the Floorplan Manager. The Floorplan Manager (FPM) is a Web Dynpro ABAP application that provides a framework for developing new Web Dynpro ABAP application interfaces consistent with SAP UI guidelines.

The entry point you need for starting an application is the application configuration, which is tied to a single Web Dynpro application. The necessary information needed to start the application is divided between the following two entities:

- **Web Dynpro ABAP Application**: Contains the information about the main component and window of the application
- **Web Dynpro ABAP Application Configuration**: Contains the information about the configuration used for starting the main component

There are only 3 different main components used in FPM-based applications. Each one corresponds to one of the supported floorplans:

- **OIF (Object Instance Floorplan)**: component FPM_OIF_COMPONENT
- **GAF (Guided Activity Floorplan)**: component FPM_GAF_COMPONENT
- **OVP (Overview Page Floorplan)**: component FPM_OVP_COMPONENT

In the screenshot below application BS_OVP_SP is using component FPM_OVP_COMPONENT. Here, it is specified that FPM_OVP_COMPONENT will start with component configuration BS_SP_OVP. As component FPM_OVP_COMPONENT is the component providing the floorplan’s functionality and layout, we will use the term ‘floorplan component’ for it and the term ‘floorplan configuration’ for the configuration used to start it.
Adaptation Options in Floorplan Manager

A Floorplan Manager UI can be adapted using different techniques. The figure below shows how the options configuration, customizing and personalization are related to each other. Context-Based-Adaptation is another way the user interface can be customized for specific use cases.

In the context of MDG you would typically choose to customize the SAP delivered configuration. Only if customizing is not feasible you would copy the SAP delivered UI configuration to the customer namespace and change the copy.

In the following cases the UI should be copied rather than customized:

- Code changes are required
- The UI needs to be changed for all users in the system and not only client specific
- The changes to the UI are extensive

Note

For more details regarding options for floorplan manager user interface adaptation, advantages, disadvantages and steps required please familiarize yourself with SAP Note 1619534
Adding a New Field to the Supplier UI for Single Maintenance (BUT000 Extension Case)

At this stage you have to make a decision whether you want to extend the standard UI configuration or create a copy. If you are only going to make small changes to the UI configuration (for example adding a field) you can follow the steps described below to extend the standard configuration. In case you need to make significant adjustments to the UI configuration it is recommended to modify a copy.

1. Run transaction SE80.
2. Open the WDR_CFGE_EDITOR Web Dynpro Component.  
   Start the customize_component Web Dynpro Application.

3. To customize the Web Dynpro component, complete the following entries and choose the New pushbutton:
   - **Component Name**: FPM_OVP_COMPONENT
   - **Configuration ID**: BS_SP_OVP

4. Enter a description.
5. Assign a transport request.
6. If you have extended the data model and you want to display additional fields on the user interface, you can now continue to make the necessary adjustments.
**Note**

If you want an overview of the UI Customizing settings or if you need to delete UI Customizing settings, refer to the following Web Dynpro applications: `WD_ANALYZE_CONFIG_USER` (user perspective), `WD_ANALYZE_CONFIG_COMP` (component perspective), and `WD_ANALYZE_CONFIG_APPL` (application perspective).

**Note**

UI customizing is client specific.

---

**Adding a New Field to the Supplier UI for Single Maintenance (Extended Field Exists in Vendor / Customer Only)**

At this stage you have to make a decision whether you want to extend the standard UI configuration or create a copy. If you are only going to make small changes to the UI configuration (for example adding a field) you can follow the steps described below to extend the standard configuration. In case you need to make significant adjustments to the UI configuration it is recommended to modify a copy.

7. Run transaction `SE80`.
8. Open the `WDR_CGFE_EDITOR Web Dynpro Component`.
Start the **customize_component Web Dynpro Application**.

9. To customize the Web Dynpro component, complete the following entries and choose the **New** pushbutton:
   - **Component Name**: FPM_OVP_COMPONENT
   - **Configuration ID**: BS_SP_OVP

10. Enter a description.
11. Assign a transport request.
12. If you have extended the data model and you want to display additional fields on the user interface, you can now continue to make the necessary adjustments.
Figure: Selecting the page **BS_SP_GENERAL_DATA** (ERP Vendor) to add the new field

Figure: Adding the new field to the **BS_SP_GENERAL_DATA** (ERP Vendor) page
Related Content

1. For more information, visit the Master Data Management homepage. (http://www.sdn.sap.com/irj/sdn/nw-mdm)
SAP How-To Guide: Extend MDG-S / MDG-C Data Model by a New Field (Reuse Option)

Copyright

© Copyright 2014 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, DB2 Universal Database, System i, System i5, System p, System p5, 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.

BusinessObjects and the BusinessObjects logo, BusinessObjects, Crystal Reports, Crystal Decisions, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of BusinessObjects S.A. in the United States and in other countries. BusinessObjects 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.