SAP How-To Guide: Extend MDG-S / MDG-C 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 an out-of-the box solution for the central management of various master data objects such as financial objects, supplier and material. However, SAP Master Data Governance also provides the flexibility to customize the solution, in cases where the pre-delivered content does not fully match customer requirements. 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 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 (i.e. Partner Functions).

Author: Lars Rueter
Company: SAP AG
Created on: March 2012
Version: 1.0
## Table of Contents

**Introduction**.................................................................................................................................................. 3
**Prerequisites**..................................................................................................................................................... 3
**Scenario**........................................................................................................................................................... 3
  - High Level Requirements.................................................................................................................................. 4
  - Governance Process........................................................................................................................................ 4
**Data Model**....................................................................................................................................................... 4
  - Storage and Use Types....................................................................................................................................... 5
  - Relationship Type............................................................................................................................................... 7
  - Reuse Area versus the Flexible Option ........................................................................................................... 7
**User Interface Configuration**.......................................................................................................................... 9
  - Adaptation options in Floorplan Manager ...................................................................................................... 10
**Implementation**.................................................................................................................................................. 11
  - Data Model Extension....................................................................................................................................... 12
    - Create a new Entity Type................................................................................................................................. 12
    - Generate MDG data model specific structures ............................................................................................. 18
**Extending the UI configuration**........................................................................................................................ 22
  - Extend the GenIL (Generic Interaction Layer) model ........................................................................................ 22
  - Connect the MDG Data Model with the GenIL Data Model ........................................................................... 27
  - Copy standard MDG-UI configurations ............................................................................................................ 29
    - Copy Application Configuration for Supplier OVP ......................................................................................... 29
    - Copy Supplier Settings for MDG Communicator............................................................................................... 30
    - Copy Supplier OVP.......................................................................................................................................... 31
  - Create List UIBB .............................................................................................................................................. 33
    - Copy List UIBB............................................................................................................................................... 33
    - Extend UIBB List Component.......................................................................................................................... 35
  - Replace OVP component in application configuration for supplier ................................................................. 39
  - Add custom list UIBB to supplier OVP............................................................................................................... 40
  - Testing the configuration.................................................................................................................................... 44
**Related Content**................................................................................................................................................ 46
**Copyright**.......................................................................................................................................................... 47
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. 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/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).

Prerequisites

You must have access to an ERP 6 EhP6 system with the MDG business functions enabled.

You must have access to the system used for cross-client maintenance.

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 entity type: “Purchasing Info Record”. The entity type should have a 1: N relationship to the Business Partner.

Figure: Data Model – Supplier (Scope of 2011 Delivery) with custom entity “Purchasing Info Record”
High Level Requirements

The business requires the new entity type called “Purchasing Info Record” as part of the MDG Business Partner data model.

Governance Process

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

Data Model

In MDG the data model is a central part of the application. SAP delivers several preconfigured data models that you can start using with little configuration. In the section High Level Requirements above you have seen the data model for Supplier (Scope of 2011 Delivery) that is shipped with MDG.

Before you start extending the data model you should familiarize yourself with some of the basic concepts. The meta-model below shows the basic elements making up a MDG data model. When you extend the data model by a new Entity Type (for example a Purchasing Info Record) you must also define its relationship to other data model elements and decide on a Storage and Use Type for the new Entity Type. See the following sections for more details regarding these topics.

Figure: The meta model is an entity-relationship-model (ERM)
The preconfigured data model for the business object types Customer / Supplier is BP. You can view the SAP delivered data model in Customizing for Master Data Governance under General Settings > Data Modeling > Edit Data Model (view cluster VC_USMD001).

### Storage and Use Types

You use storage/use type to specify whether and how master data can be changed in Master Data Governance. The storage and use type also indicates which database tables are generated by the system.

| Changeable via Change Request; Generated Database Tables | The master data of this storage and use type can be changed in Master Data Governance with a change request. The system generates all necessary database tables: check and text tables as well as additional tables, for example, for attachments and sets. The common key fields of these tables are:

- The entity type itself
- The edition – if you previously specified in the data model that the validity of master data changes is restricted to editions
- The entity types that are assigned to the entity type through leading relationships

Furthermore, all tables contain a checkbox that indicates whether the master data record is active. Depending on the workflow template used, it is possible that a master data record is not set to Active until the change request in which the record was created or changed is released.

The settings you make for the entity type (such as language dependency) result in additional key fields in the text table and the tables for attachments and sets.

The non-key fields contained in the text table are the entity texts. The non-key fields contained in the check table are the attributes of the entity type. The attachment and set tables contain predefined non-key fields. Furthermore, all database tables contain a checkbox that indicates whether the master data record was deleted. The check table also contains attributes that record which user created or changed the data records and when this was done. |
### Changeable w/o Change Request; Generated Check/Text Tables

The master data of this storage and use type can be changed in Master Data Governance without a change request. The system generates only the check and text tables with the entity type as well as with the entity types assigned to the entity type through leading relationships as fixed key fields.

The non-key fields contained in the text table are the entity texts. The check table does not contain non-key fields.

### Not Changeable via MDG; No Generated Tables

The master data of this storage and use type cannot be changed in Master Data Governance. Therefore, the system does not generate database tables. Instead, the system derives the available values from the domain that is assigned to the data element – either from the assigned value table or from the domain fixed values.

### Changeable via Other Entity Type; Generated Database Tables

The master data of this storage and use type can be changed in Master Data Governance only with a change request of an entity type with storage and use type 1. The entity type needs to be in a relationship with the relationship type leading and assigned as the To-entity type to an entity type with storage and use type 1. The system generates the check table as described for storage and use type 1, but also generates the entity types that are assigned through qualifying relationships as key fields. The system does not generate a text table, attachments, or sets since entity texts are not allowed for entity types with this storage and use type.

You can view the settings for Storage and use Type for existing Entity Types in Customizing for Master Data Governance under General Settings > Data Modeling > Edit Data Model. You select the BP Data Model and double click on **Entity Types** (view cluster VC_USMD001). In the list of entity types you can double click an entity type to view its details as shown below for Entity Type **ADDRESS**.

---

**Display View "Entity Types": Details**

![Selection box showing the different storage/use types in MDG](image)

*Figure: Selection box showing the different storage/use types in MDG*
**Relationship Type**

If you have defined multiple entity types, you can determine what type of relationship should link them (leading, referencing, qualifying, or foreign key relationship). For each relationship, you specify a relationship type and cardinality.

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referencing</td>
<td>Specifies the <em>From-Entity type</em> as an attribute of the <em>To-Entity type</em>.</td>
</tr>
<tr>
<td>Leading</td>
<td>Specifies the <em>From-Entity type</em> on a higher level than the <em>To-Entity type</em>. The <em>From-Entity type</em> is automatically taken as the key in the generated tables. A Leading relationship type is identical to a Qualifying relationship type, except when the <em>To-Entity type</em> has a Storage and Use Type of 4. Master data for <em>To-Entity types</em> in Leading relationships is processed in the context of the entity type that is assigned using the leading relationship.</td>
</tr>
<tr>
<td>Qualifying</td>
<td>Specifies the <em>From-Entity type</em> on a higher level than the <em>To-Entity type</em>. The <em>From-Entity type</em> is automatically taken as the key in the generated tables.</td>
</tr>
</tbody>
</table>

**Cardinality**

The following options are possible for the relationship between two entity types:

- **1:N**
  This cardinality represents a mandatory relationship in which one or more *To-Entity Types* can be assigned to a *From-Entity Type*.
  This cardinality is valid for relationships with the relationship types Leading, Qualifying, and Referencing. In relationships with the relationship type referencing, the *From-Entity Type* is a required attribute of the *To-Entity Type*.

- **0:N**
  This cardinality represents an optional relationship in which any number *To-Entity Types* can be assigned to a *From-Entity Type*.
  This cardinality is valid only for relationships with the relationship type Referencing. The *From-Entity Type* is an optional attribute of the *To-Entity Type*.

**Note**

Which relationship types are permitted depends on the storage and use types of the entity types ([help.sap.com](http://help.sap.com)).

**Important**

The general design assumption is that there is a 1: N relationship between a database table and its entity types. This means one entity type does not bundle several database tables.

**Reuse Area versus the Flexible Option**

When you extend the SAP delivered data model by adding a new entity type, you have to decide where to store data after activating the change request. During the processing of the change request, the data is
stored in the MDG staging area. After activation, the data can be moved to tables outside of MDG or it can stay in the MDG tables (flex option).

Where the data is stored is specified by the **Reuse Area** setting on an **Entity Type** level as shown in the screenshots below.

---

**Figure: Assignment of Reuse Area for the Data Model BP**

---

**Figure: Assignment of Reuse Area for the Entity Type BP_HEADER**
The MDG model BP is preconfigured with one reuse area called PARTNER. This reuse area points to the access class CL_MDG_BS_BP_ACCESS_MASTER, which can handle all fields of the pre-delivered SAP Business Partner.

If you extend the Data Model by a new Entity Type and want the data of that Entity Type to remain in the MDG tables after activation you can choose MDG as a reuse area.

In the figure below, the architecture behind both persistence options is shown. On the left side, the flexibility option uses Master Data Governance as persistence. The reuse option on the right side uses existing ERP tables for persistence.

#### 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.

For each main component used in FPM-based applications there is a corresponding supported floorplan. The floorplans and their components are as follows:

- **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 configuration 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 typically **customize** the SAP delivered configuration. Only if customizing is not feasible do you copy the SAP delivered UI configuration to the customer namespace and change the copy.

In the following cases, we recommend you copy rather than customize the UI:

- 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
Implementation

Two major building blocks make up the implementation of the Entity Type extension. In the first phase, the MDG data model is extended. In the second phase, the User Interface is extended to include the new entity. The flow diagram below shows the detailed implementation steps and should be used as an orientation. Each box in the diagram below corresponds to a section in this guide where you find detailed execution instructions.

![Flow diagram](image)

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

You want to extend the MDG data model for Business Partner by an additional entity “Purchasing Info Record”. The entity should have a 1: N relationship to 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 Diagram]

*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.
12. A messages popup is displayed (see screenshot.)

13. Choose the **Adjust Staging Area** as shown to adjust existing change requests.

14. In the following steps, you verify if the MDG staging structures where successfully generated.

   Choose the **Visualize Data Model** pushbutton.

15. Choose the **Active Version** pushbutton.

16. In the screenshot, you see the generated structures.
17. To view the generated tables, start transaction SE38.
Enter program USMD_DATA_MODEL.
Enter data model BP.
Run the program.

![ABAP Editor: Initial Screen](image)

18. Double click on ZINFOREC.

![Generated Tables](image)

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

![Image of a generated table](image)

### Generate MDG data model specific structures

Since the MDG data model was changed you need to 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.
### 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.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20.</td>
<td>Log into system for cross-client maintenance.</td>
</tr>
</tbody>
</table>
| 21. | Start Customizing for **Master Data Governance** (transaction `MDGIMG`).  
  Go to **General Settings -> Data Modeling -> Generate Data Model-specific structure** |
22. Create two New Entries as shown in the screenshot.

One entry for Active Area Mapping.
On entry for Field Properties.
Save your changes.

23. Select the row with data model BP

Double-click Structures in the left hand panel.
Click Generate Structures.

24. Verify that your structures for ZINFOREC were successfully generated.
25. In the following steps you verify that the active area mapping structure was successfully generated.

Start transaction **SE11**.

Display structure **ZXX_S_BP_FF_ZINFOREC** by entering the details as shown.

26. You have now verified that the structure has been generated.
Extending the UI configuration

Extend the GenIL (Generic Interaction Layer) model

27. Create an enhancement for the standard BUPA GenIL Model (transaction code: genil_model_browser)

28. Enter **Superenhancement**: BUPA_CUSP.
29. Verify the details of the enhancement.

<table>
<thead>
<tr>
<th>Component Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhancement</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Implementation Class</td>
</tr>
<tr>
<td>Superenhancement</td>
</tr>
<tr>
<td>Prefix</td>
</tr>
<tr>
<td>Final</td>
</tr>
<tr>
<td>Created</td>
</tr>
<tr>
<td>Last Change</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Implementation Class</td>
</tr>
<tr>
<td>Object Table</td>
</tr>
</tbody>
</table>

30. Save the enhancement.

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

33. 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).
34. 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.

35. Verify that the model nodes look similar to the ones shown.
36. Switch to edit mode. Navigate to the **Relations** node. Right-click and select **Create Relations**.

37. Enter the details as shown.
38. In the relations detail screen select the assigned object **ZINFOREC**.

<table>
<thead>
<tr>
<th>Module</th>
<th>Table Title</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation</td>
<td>Enhancement</td>
<td>Original Definition In: ZINFOREC, Displayed Definition In: ZINFOREC</td>
</tr>
<tr>
<td>Basic Settings</td>
<td>Component</td>
<td>BUPA</td>
</tr>
<tr>
<td></td>
<td>Relation</td>
<td>Zop2zinfo</td>
</tr>
<tr>
<td></td>
<td>Source Object</td>
<td>BP_Root</td>
</tr>
<tr>
<td></td>
<td>Card. of Src Object</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Relation Type</td>
<td>Aggregation</td>
</tr>
<tr>
<td></td>
<td>Card. of the Assigned Object</td>
<td>0, 0</td>
</tr>
<tr>
<td></td>
<td>Assigned Object</td>
<td>ZINFOREC</td>
</tr>
</tbody>
</table>

39. Save your changes.

**Connect the MDG Data Model with the GenIL Data Model**

Make entries in the view **VC_MDG_BS_GENIL_C** to relate the GENIL model component to the MDG data model entity type.
<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.</td>
<td>Start transaction SE34. Display View Cluster VC_MDG_BS_GENIL_C.</td>
</tr>
<tr>
<td>41.</td>
<td>Choose the <strong>New Entries</strong> pushbutton.</td>
</tr>
</tbody>
</table>
| 42.  | Create a new entry with the following details:  
  - **Comp. Name**: BUPA  
  - **Object Type**: BUS1006  
  - **External Object**: BP_Root |
| 43.  | Mark the new entry and select **Comp – GenIL Node**. Then choose the **New Entries** pushbutton. |
| 44.  | 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 OVP

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

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

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

48. **Copy the existing configuration to ZBS_OVP_SP.**
   
   It is important that you use the same **Name** as for 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

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

50. Choose the **Start Configurator** pushbutton.
51. Copy the existing configuration to ZZBS_SP_OVP.

52. Refresh the list of configurations and look for your copy. It should be there.
53. Start transaction SE80.
   Navigate to the **Component Configuration** as shown.

54. Locate the **FPM_LIST_UIBB TEMPLATE** in the list of component configurations.
   Choose the **Start Configurator** pushbutton.
55. Copy the existing configuration to ZFPM_LIST UIBB PIR.

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

57. Start transaction **SE80**.

   Navigate to the *Component Configuration* as shown.

58. Locate your own **ZFPMLIST_UIBB_PIR** configuration in the list of component configurations.
59. Choose the **Start Configurator** pushbutton.

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

61. Choose the **Feeder Class** pushbutton and enter the feeder class as shown.
62. Enter the feeder class parameters as shown.

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

64. Add your fields as table columns as shown.
65. Save your changes.

66. Nothing needs to be added to the **Toolbar Schema** tab page.
Replace OVP component in application configuration for supplier

67. Start transaction SE80.

   Navigate to your Application Configuration as shown.

68. Choose the Continue in Change Mode pushbutton.

69. Select the row starting with OVP and choose the pushbutton Assign Configuration Name.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>70.</td>
<td>Change the configuration name as shown.</td>
</tr>
<tr>
<td>71.</td>
<td>Save your changes. Click on the link ZZZP_SP_OVP to edit the configuration.</td>
</tr>
</tbody>
</table>

**Add custom list UIBB to supplier OVP**

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>72.</td>
<td>Open the pushbutton <strong>UIBB</strong> menu, as shown.</td>
</tr>
</tbody>
</table>
73. Select **Freestyle Component** from the drop down as shown.

74. The empty property list appears.

75. Enter the attribute values as shown.
76. Choose the **Toolbar Schema** tab.

77. 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.
78. Enter the attribute values as shown.

![Diagram showing attribute values]

79. 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
80. Change to the Wire Schema tab.
Choose the **Wire** pushbutton.

81. 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:** *
- **Connector Parameters**
  - **Relation Name:** zbp2inforec
  - **Creation Mode:** Creation with Default Values

**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
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, DB2 Universal Database, System i, System i5, System p, System p5, System x, 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, Nettivity, 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.