How To... Master Data Governance for Material: Create Custom Print forms

Applicable Releases:
MDG 7

Version 1
January 2014
<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>First official release of this guide</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1. **BUSINESS SCENARIO** ................................................................................................................................. 4

2. **BACKGROUND INFORMATION** .................................................................................................................. 5

3. **STEP BY STEP EXPLANATION** ...................................................................................................................... 6

   3.1. **PREREQUISITE** ............................................................................................................................................. 6

   3.2. **COMPONENTS** ............................................................................................................................................ 6

   3.3. **CREATE NEW PRINT FORM** ...................................................................................................................... 6

   3.4. **CREATE NEW INTERFACE** ...................................................................................................................... 7

   3.5. **CHANGE LOGO** ........................................................................................................................................... 8

   3.6. **ADD CUSTOM FIELD** ............................................................................................................................ 8

   3.7. **DELETE FIELD** ........................................................................................................................................... 9

   3.8. **MULTIPLE USAGE OF A FIELD** ................................................................................................................ 9

   3.9. **PLACEHOLDER IN HEADER LINE** ........................................................................................................ 10

   3.10. **FIELD TEMPLATES** .................................................................................................................................. 10

   3.11. **CUSTOMIZING: ASSIGN NEW FORM TO UI CONFIGURATION** ............................................................. 11

   3.12. **ERROR ANALYSIS** .................................................................................................................................... 12
1. BUSINESS SCENARIO

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

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

With MDG 7 SAP delivers a new print form, MDG_BS_MAT_PDF_04, in package MDG_BS_MAT_PDF.

This guide shows you how to create a custom print form to support different layouts and custom fields.
2. BACKGROUND INFORMATION
MDG-M uses Adobe LiveCycle Designer to create forms used to print material master data.

There is a lot of information on the SAP Community Network (http://scn.sap.com) and at http://www.adobe.com.

SAP also provides a course with further information on print forms, BC480 (PDF-based Print Forms).
3. **STEP BY STEP EXPLANATION**

The following explanation shows how to create a custom print form to support different layouts and custom fields.

### 3.1. Prerequisite

To be able to edit the forms delivered by SAP you have to install Adobe LiveCycle Designer on your computer. SAP Note 1522483 describes how to get and install the tool. Also you may want view the SAP help portal [http://help.sap.com](http://help.sap.com) and search with keywords “Adobe LiveCycle Designer Installation” to get the relevant information on how to install and work with Adobe LiveCycle Designer. After installation you'll find a folder (named similar to C:\Program Files (x86)\Adobe\Designer 9.0\Documentation) on your hard disk with offline documentation in different languages (for example, subfolder EN for English). The file “Designer.chm” is opened when pressing the key F1 within the form layout editor.

### 3.2. Components

Four components are involved in printing:

- A UI with a print button that creates an event when clicked
- A print program/method (CL_MDG_BS_MAT_PDF=>PRINT_MATERIAL) that is called by the event handler (CL_MDG_BS_MAT_APPCC=>/PLMU/IF_EX_FRW_APPCC_OVP~PROCESS_EVENT) to read the data to be printed
- A form interface (MDG_BS_MAT_PDF) that defines the data that can be added to the layout of the form
- A print form (MDG_BS_MAT_PDF_04) that defines the layout of the form

### 3.3. Create new Print Form

SAP recommends not to change the delivered form but to make a copy and to change the copy. The copied form can easily be assigned within the customizing to be used by the application (see chapter 3.11).

Execute the following IMG activity:

![Display IMG](image)

You may also want to start transaction SFP directly.
Copy the SAP-delivered print form MDG_BS_MAT_PDF_04 to a new one:

3.4. Create new Interface
If you intend to pass additional or different data to the form you have to copy the interface as well:

Afterwards you have to assign the copied interface to the copied form:

Data is passed to the print form using parameter IS_DATA. It is of type MDG_BS_MAT_S_MAT_PDF that can be enhanced deeply, so it is generally not necessary to exchange the interface. Only if you want to pass additional parameters or custom entities you need to do so. But be aware that you then also have to modify the print method
CL_MDG_BS_MAT_PDF=>PRINT and maybe also CL_MDG_BS_MAT_PDF=>PRINT_MATERIAL because these methods are currently not designed for customer extensions.

3.5. Change Logo

The logo at the right upper corner of the form has been defined on the master page within sub form “Header”. It can be replaced by an own logo by specifying the URL of the logo.

If the logo should be embedded into the form, option “Embed Image Data” needs to be checked.

3.6. Add Custom Field

After extension of a database table (for example, MARA) the corresponding structure (for example, MDG_BS_MAT_S_MARA) already contains the new fields and the only thing to do is a refresh of the corresponding context element within the form context.

Afterwards the new fields are available within the data view of the layout.
The text of a field value (for example, text Kilogram for value KG) is stored in a separate sub structure for each table structure. It is called USMD_DESCRIPTION and has to be enhanced by a customer include with a field of type USMD_TXTLG. The field name has to be the same as the data field (for example, GEWEI for weight unit) within the corresponding table structure. The text of the field value should be determined automatically because of the text table reference definition within the data dictionary. In special cases it might be necessary to adapt this text field value determination. It is located in method CL_MDG_BS_MAT_TEXT->GET_TEXT_FIELD_VALUE.

3.7. Delete Field

A field that should not be displayed can be deleted easily by selecting the field within the design view or the hierarchy view and pressing the delete key.

3.8. Multiple Usage of a Field

It is not possible to display a field value at multiple places by defining the data binding to the same context field.

The data binding of the corresponding fields can be defined as *Use global data*. All fields with this name will then have the same value. This may be risky if a field name has been used within different structures and the field values are not the same at runtime. Another problem is that the usage of such fields is not visible within the data view.

*Use global data* has been used with field *Change Request ID* (CR_ID) that is in sub form *Footer* of the master page and on page *ChangeRequest* in sub form *GeneralData*.

Another possibility is using scripting.

**Example:**

Field GEWEI is used twice on page *Material*, sub form *BasicData*, sub form *Dimension*. The second usage has no data binding as it is defined at the first field ($MARA_TAB.DATA[*].GEWEI), but a script at event *form.ready*:

```
$.rawValue = $.parent.GEWEI.rawValue
```
The value of the second field \textit{GEWEI} references to the value of field \textit{GEWEI} that is within the current sub form (specified by parent).

If a reference to a field on a different page or sub form is required, \texttt{.parent} has to be used for each upper level within the object hierarchy up to a common node. Navigation down the object hierarchy to the required field has to be done by specifying the names of the pages and/or sub forms.

3.9. \textbf{Placeholder in Header Line}

Scripting has to be used to replace placeholders (for example, \texttt{&LGNUM&} within header line \textit{Warehouse \&LGNUM\&: Data}) in text values. A special hidden page, \texttt{Scripts_Deactivated_NotVisible}, has been created to replace such placeholders at a central place.

Example:

\begin{verbatim}
Material.WarehouseLGNUMData.Heading =
Replace(Material.WarehouseLGNUMData.Heading.rawValue, "&LGNUM&",
$.CURRENT_LGNUM.rawValue)
\end{verbatim}

At event, \texttt{form:ready} of page \texttt{Scripts_Deactivated_NotVisible} replaces \texttt{&LGNUM&} within above specified example header line, that is on page \texttt{Material} within sub form \texttt{WarehouseLGNUMData} and there in text field \textit{Heading}. The replaced value is stored in field \texttt{CURRENT_LGNUM} that is on the current page and has a data binding to context structure \textit{HEADER}, field \texttt{CURRENT_LGNUM}.

3.10. \textbf{Field templates}

When dropping a field from the data view to the layout the field gets a default layout. If the layout does not meet the requirements, it must be changed for each field separately.

When a field layout has been finally changed it can be stored as template.

Then it can be used for further fields with the same layout. The layout should be stored without binding. It may lead to confusion if it is not changed properly after using the template. Also other layout elements like header lines or groups of objects can be stored as template.
3.11. Customizing: Assign new Form to UI Configuration

To assign your print form to your UI configurations use the following IMG activity:

If no form for the current UI configuration can be found message MDG_BS_MAT_PDF 015 is displayed.
3.12. Error analysis

There are several causes why a field value may be not visible on a form. The following list provides some hints on what to check:

1. Be sure that the changed form has been activated and that the MDG-M session has been restarted. Otherwise the old but active form and its function module (see 7.) are used.

2. Data to be printed is read in method CL_MDG_BS_MAT_PDF=>PRINT_MATERIAL. Check there whether the corresponding entity data has been read.

3. Entity data is mapped to its corresponding table fields of structure MDG_BS_MAT_S_MAT_DATA in the methods called for each entity by CL_MDG_BS_MAT_PDF=>PRINT_MATERIAL. Method CL_MDG_BS_MAT_SERVICES=>MAP_MODEL_2_API does the data transfer from entity or model structure to API structure. The transfer to the form structure MDG_BS_MAT_S_MAT_PDF is done by method CL_ABAP_TOOL=>MOVE_COMPLEX whenever necessary.

4. As already mentioned within chapter 3.6 the text value of a field value should be determined automatically within method CL_MDG_BS_MAT_PDF=>SET_DESCRIPTION. A prerequisite is that the text field has been defined within the sub structure USMD_DESCRIPTION with the same name as the data field and that a corresponding data field exists.

5. If the same field has been used multiple times, be sure to have taken into account the hints from chapter 3.8.

6. Check whether the data binding of the field to the context field within the form layout is correct. You may also want to check whether a field has a data binding by checking the usage icon on the right side of a field within the data view of the form layout.

7. Each activated form has a corresponding generated function module. The name of the function module is displayed when testing the form. You also can set a breakpoint within function module FP_FUNCTION_MODULE_NAME to get the corresponding function module name of the form.