

SAP
How-To Guide
Master Data Governance
for Material



How To...
Master Data Governance for Material:
UI Badi for MDG Material
(BADI USMD_UI_EVENT2)

Applicable Releases:

EhP5

Version 1.0

August 2011

© Copyright 2011 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, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

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 Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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 in several other countries all over the world. 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.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Document History

Document Version	Description
1.00	First official release of this guide

Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1. Business Scenario	2
2. Background Information	2
3. Prerequisites	2
4. Step-by-Step Procedure	3
4.1 Hide Tabs/GUIBBs	3
4.1.1 Tabs should not be visible	3
4.1.2 Hide tabs depending on role	3
4.2 Only specific roles are allowed to change data	4
4.2.1 Role specific display for classification data.....	4
4.2.2 Role specific display of fields	5
4.2.3 Role specific hiding of Buttons.....	8
4.3 Add new Button and process its event	10
4.4 Check if deletion of a list entry (e.g. EAN) is allowed.....	10
4.5 Initialize fields	10
4.6 Handle field properties dynamically	10
4.7 Restrict drop down list	10
4.8 Restrict F4 value help	10
4.9 Control changes of row selections in tables	10
4.10 Workflow step dependent properties	10

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 in a central hub and the distribution to connected operational and business intelligence systems.

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

This scenario addresses the customer requirement to have a more flexible user interface. This could be dependent for example on roles.

This How To Guide shows some examples to adapt the UI with the BADI `USMD_UI_EVENT2`.

You can use this BADI to modify the user interface in single processing for material.

You can customize the user interface as follows:

- Adjust the definition of attributes or add new attributes
- Initialize the displayed data (when creating a new entity, for example)
- Restrict the values displayed in a dropdown box or selection field group
- Restrict the values displayed in the input help
- Dynamically control the visibility of fields on the user interface and of the property that determines if fields are required or display-only
- Define navigation destinations of UI elements of the type hyperlink (or pushbutton)
- Check if the lead selection of a table may be changed

2. Background Information

Considered for Field properties/Authorities for Material UI:

- Field properties Backend (via T130F and BADI_MAT_F_SPEC_SEL) considered via API
- Authorities Backend (Security Guide) considered via API
- UI-BADI (da macht der Kunde, was er will)

Not considered:

- authorization check in Modell MM ('Define Authorization Relevance per Entity Type')
- 'Required Field' flag on attribute level

3. Prerequisites

- Knowledge of ABAP-OO, ABAP Webdynpro (WD) and Floor Plan Manager (FPM)
- Configuration Guide:
http://help.sap.com/erp2005_ehp_05/helpdata/en/b2/9a55fd374848a1b70679180b08514b/frameaset.htm
- Central guide for SAP MDG-M extensibility:
<http://www.sdn.sap.com/irj/sdn/master-data-governance?rid=/webcontent/uuid/f0801486-9562-2e10-54b1-9c123497f1a7> (Chapter UI BADI)

4. Step-by-Step Procedure

4.1 Hide Tabs/GUIBBs

4.1.1 Tabs should not be visible

IF_EX_USMD_UI_EVENT2~MODIFY_DEFINITION

Example: The Tabs “Basic Text” and “Internal Comment” should not be visible at all (this might of course also be realized within Web-Dynpro configuration) .

Method: IF_EX_USMD_UI_EVENT2~MODIFY_DEFINITION

```
* MM_BASIC_TEXT
ls_hide_view-mainview = 'MM_BASIC_TEXT'.
INSERT ls_hide_view INTO TABLE et_hide_view.
* internal comment
ls_hide_view-mainview = 'MM_INT_COMMENT'.
INSERT ls_hide_view INTO TABLE et_hide_view.
```

4.1.2 Hide tabs depending on role

A custom table is defined where a user is assigned to a role, which in this example is represented by a two digit numerical character (type xrole), which is determined within the function module ‘ZGET_ROLE_FROM_USER’. Of course the role definition might also be derived directly from the user master data.

Method: IF_EX_USMD_UI_EVENT2~MODIFY_DEFINITION

```
data: lv_role      type xrole,
      ls_hide_view type usmd_s_ui_view.
* Hide tabs depending on role
call function 'ZGET_ROLE_FROM_USER'
  exporting i_user      = sy-uname
  importing ev_role     = lv_role
  exceptions
    usr_assignment = 1
    others         = 2.
if sy-subrc <> 0.
  exit.
endif.
case lv_role.
  when '00'. " Role 00
*   MRP
  ls_hide_view-mainview = 'MAINVIEW_9'.
  insert ls_hide_view into table et_hide_view.
*   Sales
  ls_hide_view-mainview = 'MAINVIEW_10'.
  insert ls_hide_view into table et_hide_view.
*   Tax
  ls_hide_view-mainview = 'MAINVIEW_11'.
```

```

        insert ls_hide_view into table et_hide_view.

when '02'. " Role 02
*   Sales Data
    ls_hide_view-mainview = 'MAINVIEW_10'.
    insert ls_hide_view into table et_hide_view.
*   Tax
    ls_hide_view-mainview = 'MAINVIEW_11'.
    insert ls_hide_view into table et_hide_view.
endcase.

```

4.2 Only specific roles are allowed to change data

Use IF_EX_USMD_UI_EVENT2~MODIFY_VIEW: method to dynamically adjust the UI

4.2.1 Role specific display for classification data

The following example dynamically uses again the function module ZGET_ROLE_FROM_USER to correct (=take away) role-dependent the input property from classification and other data.

Insert the following source code into the existing original. Its effect will be that the classification data will not be changeable (display only),

```

*   Now the context for lead object can be changed
*   -----
INSERT:
*   Only Roles 00 and Role 01 are allowed to change class/characteristics

data: lv_role type xcrole.
      call function 'ZGET_ROLE_FROM_USER'
          exporting
              i_user          = sy-uname
          importing
              ev_role         = lv_role
          exceptions
              usr_assignment = 1
              others         = 2.

      if not ( lv_role eq '00' or lv_role eq '01' ).
          lv_ui_mode = 'D'.
      endif.
endif.

END OF INSERT:

      TRY.
          CALL METHOD lo_classification->set_context
              EXPORTING
                  iv_objtype      = if_mdg_bs_mat_gen_c=>gc_objtype_mara
                  ...

```

4.2.2 Role specific display of fields

In this chapter the input readiness is restricted for certain roles within a newly created method SET_FIELDS. So, whenever this method is left due to a check or any other condition without proceeding to the end all fields (of course depending on the basic configuration) are open for input.

Again you use IF_EX_USMD_UI_EVENT2~MODIFY_VIEW.

The call of the new method SET_FIELDS should be placed at the end of the standard section of MODIFY_VIEW (for this example only a reduced set of the interface is in use). The export parameters are already known (from the interface of MODIFY_VIEW).

** Field Properties*

```
CALL METHOD me->set_fields
  EXPORTING
    is_ui_context = is_ui_context
    it_key_field  = it_key_field
  IMPORTING
    et_property  = et_property
    et_fpm_event = et_fpm_event
  CHANGING
    ct_data      = ct_data

ENDMETHOD.
```

The method SET_FIELDS has the signature shown in the screen shot below; for simplicity the interface of MODIFY_VIEW was just copied.

▶	IS_UI_CONTEXT	TYPE USMD_S_UI_CONTEXT2	UI Context
▶	IT_KEY_FIELD	TYPE USMD_TS_VALUE	Field Name and Value
▶	IO_FPM_EVENT	TYPE REF TO CL_FPM_EVENT OPTIONAL	CL_FPM_EVENT
▶	ET_PROPERTY	TYPE USMD_T_UI_PROPERTY2	UI Properties
▶	ET_FPM_EVENT	TYPE USMD_TS_FPM_EVENT	FPM Event
▶	ET_MESSAGE	TYPE USMD_T_MESSAGE	Messages
▶	CT_DATA	TYPE USMD_TS_ENTITY_DATA	UI Data
▶	CD_LEAD_INDEX	TYPE SYTABIX OPTIONAL	Index of Internal Tables

The table ET_PROPERTY contains the information how the corresponding fields will be output finally. The following section highlights the essential statements:

```
ls_property-attribute = lv_field_name.
ls_property-rownr    = lv_index.
*
ls_property-required = ''.
ls_property-read_only = 'X'.
ls_property-visible  = 'X'.
```

```
insert ls_property into table et_property.
```

As an example the method SET_FIELDS would contain following statements:

A further function module ZGET_FIELDS_FROM_ROLE is introduced which fulfills a double purpose, firstly, it returns – just like the function module ZGET_ROLE_FROM_USER above – the role of the current user as a two digit numerical field and then, secondly, the list of maintainable fields for this specific user role (contained in table lt_fields; the connected table type zfields contains at least the field table name, value for instance 'MARA', and a field name).

The meaning of the checks below is an arbitrary example; the working principle is important, however, all other shown code should be copied or coded similarly to ensure a smooth processing (steps are needed to find out the name of the UIBB currently in work and to handle the field properties within it).

One note in addition: the statements have been copied from an existing example which served further purposes, so the sequence might be changed for optimization if only this example were the scope.

```
method SET_FIELDS.
  data: lv_structure type ddojname,
        lr_line type ref to data,
        lr_line2 type ref to data,
        lr_table type ref to data,
        lv_error type ref to cx_root,
        struct_descr type ref to cl_abap_structdescr,
        lv_field_name type string,
        ls_property type usmd_s_ui_property2,
        lv_table type string,
        lt_fields type zfields,
        lv_index type sy-tabix,
        lv_role type xcrole,
        ls_fpm_event type usmd_s_fpm_event.

  field-symbols: <fs_fields> type xzfields.

  field-symbols: <fs_line> type table,
                 <fs_table> type table,
                 <comp_descr> like line of cl_abap_structdescr=>components,
                 <fs_selected_field> type any,
                 <fs_data> type usmd_sx_entity_data.

  read table ct_data assigning <fs_data> index 1.
  try.
    * get the structure definitions and data references assigned
    assign <fs_data>-r_data->* to <fs_line>.
    create data lr_line like line of <fs_line>.
    struct_descr ?= cl_abap_typedescr=>describe_by_data_ref( lr_line ).
    catch cx_sy_create_data_error into lv_error.
    catch cx_sy_move_cast_error.
  endtry.

  call function 'ZGET_FIELDS_FROM_ROLE'
    exporting
```

```

    i_user          = sy-uname
importing
    et_fields      = lt_fields
    ev_role        = lv_role
exceptions
    usr_assignment = 1
    others         = 2.
if sy-subrc <> 0.
    exit.
endif.

```

* Role 00 and Role 01 do not need any additional restrictions

```
check not ( lv_role eq '00' or lv_role eq '01' ).
```

* Assign table name

```

if not is_ui_context-entity_cont is initial.
    if is_ui_context-entity_cont cp 'XZ*'.
        lv_table = is_ui_context-entity_cont.
        shift lv_table by 2 places .
    elseif is_ui_context-entity_cont = 'UNITOFMSR'.
        lv_table = 'MARA'.           "UoM like Basic view not MARM
    elseif is_ui_context-entity_cont = 'MEAN_GTIN'.
        lv_table = 'MEAN'.
    else.
        lv_table = is_ui_context-entity_cont.
    endif.
else.
    lv_table = 'MARA'.
endif.

```

* How many lines? At least one time

```

lv_index = lines( <fs_line> ).
if lv_index = 0.
    lv_index = 1.
endif.

```

```
do lv_index times.
```

* Table MARA read-only for all roles except Role 00 and 01; checked above

* MLAN editable only for Role 06.

```

    check lv_table eq 'MARA'           or
           lv_role eq '03'             or
           lv_role eq '05'             or
           ( lv_role eq '02' and lv_table ne 'MARC' ) or
           ( lv_role eq '04' and lv_table ne 'MVKE' ) or
           ( lv_role eq '06' and lv_table ne 'MLAN' ).

```

* loop at fields and assign value of selected data

```

loop at struct_descr->components assigning <comp_descr>.
    lv_field_name = <comp_descr>-name.
    read table lt_fields assigning <fs_fields>

```

```

        with key xctable = lv_table
            xcfield = lv_field_name.

*   lt_fields contains maintainable fields
    if sy-subrc <> 0.
        ls_property-attribute = lv_field_name.
        ls_property-rownr    = sy-index.
*   ls_property-required = ''.
        ls_property-read_only = 'X'.
        ls_property-visible = 'X'.
        insert ls_property into table et_property.
    endif.
endloop.
enddo.

```

4.2.3 Role specific hiding of Buttons

Use IF_EX_USMD_UI_EVENT2~MODIFY_VIEW

This section could run after the role dependent field control settings or in any section where the role of the current user (here lv_role) is known. In addition you have to make sure that the used interface parameters like iv_entity (i.e. the entity) and the Floorplan-Manager event table (here et_fpm_event) have been defined.

A new method SET_BUTTONS should be created. It appends lines to table et_fpm_event which contain solely the name of the concerned entity within the data model (e.g. XZMVKE) and names of the buttons that have to be taken away for display-only roles. Only in this case nothing else has to be specified. The further options are contained within table type usmd_s_fpm_event.

The three possible values for add line, copy line and delete line are listed as comment in method SET_BUTTONS. Only exception is the tab for unit of measure. Here you have to specify another delete event because this tab also has dependencies to the basic data of the material and requires an individual processing.

```

method SET_BUTTONS.
    data: ls_fpm_event type usmd_s_fpm_event.

*   USMD_TABLE_ADD_LINE
*   USMD_TABLE_COPY_LINE
*   USMD_TABLE_DEL_LINE

    ls_fpm_event-entity      = iv_entity.
    ls_fpm_event-eventid    = 'USMD_TABLE_ADD_LINE'.
    insert ls_fpm_event into table et_fpm_event.

    ls_fpm_event-entity      = iv_entity.
    ls_fpm_event-eventid    = 'USMD_TABLE_COPY_LINE'.
    insert ls_fpm_event into table et_fpm_event.

    if iv_entity = 'UNITOFMSR'.
        ls_fpm_event-entity      = iv_entity.

```

```

ls_fpm_event-eventid    = 'EVENT_DEL_UOM'.
insert ls_fpm_event into table et_fpm_event.
else.
ls_fpm_event-entity     = iv_entity.
ls_fpm_event-eventid    = 'USMD_TABLE_DEL_LINE'.
insert ls_fpm_event into table et_fpm_event.
endif.
endmethod.

```

Disable add/copy/delete buttons above a table UIBB like basic text, EAN, or any other view (for instance plant). In the example below the buttons above the plant view are disabled role dependent, all other always

```

* Basic data: disable for all roles

* EAN
call method me->set_buttons
  exporting
    iv_entity    = 'MEAN_GTIN'
  importing
    et_fpm_event = et_fpm_event.

* Basic Text
call method me->set_buttons
  exporting
    iv_entity    = 'BSCDATTXT'
  importing
    et_fpm_event = et_fpm_event.

* role specific changes
case lv_role.

  when '03' or '04'.
*   hide MARC buttons for 03 and 04
  call method me->set_buttons
    exporting
      iv_entity    = 'XZMARC'
    importing
      et_fpm_event = et_fpm_event.

  when '05'.
*   hide MARC, MVKE, MLAN buttons for 05
  call method me->set_buttons
    exporting
      iv_entity    = 'XZMVKE'
    importing
      et_fpm_event = et_fpm_event.

  call method me->set_buttons
    exporting
      iv_entity    = 'XZMARC'

```

```
importing
    et_fpm_event = et_fpm_event.

call method me->set_buttons
exporting
    iv_entity    = 'XZMLAN'
importing
    et_fpm_event = et_fpm_event.
endcase.
```

4.3 Add new Button and process its event

IF_EX_USMD_UI_EVENT2~MODIFY_DEFINITION

IF_EX_USMD_UI_EVENT2~PROCESS_EVENT

4.4 Check if deletion of a list entry (e.g. EAN) is allowed

Check if deletion of a list entry (e.g. EAN) is allowed and raise a message if necessary message

IF_EX_USMD_UI_EVENT2~MODIFY_VIEW

IF_EX_USMD_UI_EVENT2~PROCESS_EVENT

4.5 Initialize fields

IF_EX_USMD_UI_EVENT2~INITIALIZE

IF_EX_USMD_UI_EVENT2~MODIFY_VIEW

4.6 Handle field properties dynamically

IF_EX_USMD_UI_EVENT2~MODIFY_VIEW

4.7 Restrict drop down list

IF_EX_USMD_UI_EVENT2~GET_ATTR_VALUE_SET

4.8 Restrict F4 value help

IF_EX_USMD_UI_EVENT2~PROCESS_VALUE_HELP

4.9 Control changes of row selections in tables

IF_EX_USMD_UI_EVENT2~LEAD_SELECTION_CHANGED

4.10 Workflow step dependent properties

IF_EX_USMD_UI_EVENT2~MODIFY_VIEW

IF_EX_USMD_UI_EVENT2~SET_MODE_TEXT_TRANSL

www.sdn.sap.com/irj/sdn/howtoguides