



PUBLIC

# Extend the Business Partner – Custom Handler Class

Applicable Releases:  
All

Version 1.2

July 2020



## Document History

Document Version	Description
1.0	First official release of this guide with MDG 6.0
1.1	Updates for MDG 6.1
1.2	Updates for MDG 1909

1. BUSINESS SCENARIO .....	4
2. SAP NOTES AND LINKS.....	5
3. BACKGROUND INFORMATION.....	6
4. STEP BY STEP EXPLANATION.....	7
4.1. CREATE A CUSTOM HANDLER CLASS .....	7
4.2. REGISTER YOUR CUSTOM HANDLER CLASS .....	7
4.3. VERIFY THE USAGE OF YOUR CUSTOM HANDLER CLASS .....	7
4.4. ADD CUSTOM LOGIC TO YOUR CUSTOM HANDLER CLASS.....	7
4.4.1. <i>Sequence of handler class processing</i> .....	7
4.4.2. <i>Copying SAP delivered code</i> .....	8
5. PERFORMANCE.....	9

## 1. BUSINESS SCENARIO

SAP Master Data Governance for Business Partners, Customers or Suppliers (MDG-BP/C/S) provides business processes to find, and maintain business partner, customer or supplier master data. It supports data governance 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 how to guide provides an example for extending MDG-C or MDG-S by creating and registering a custom handler class.

The SAP predefined handler classes do not meet the requirements of your processes. You want to create a custom handler class for use within the MDG application layer. The given scenario is a generic description. Custom handler classes might be relevant for different scenarios.

## 2. SAP NOTES AND LINKS

In addition to the detailed explanations written in this document, please see the following SAP Notes and links for further important information:

- [1637249](#) MDG: Information for efficient message processing
  - [2105467](#) MDG Performance
  - [2221398](#) MDG-BP/C/S/CA: (Un-)Supported Fields in Data Model BP
  - [2561461](#) Scope of support for SAP Master Data Governance (MDG)
  - [2847807](#) MDG-BP/C/S/CA: Usage of MDG Tools and Processes
- [Configuration and Enhancement of SAP Master Data Governance](#)

### 3. BACKGROUND INFORMATION

Custom handler classes must implement the interface `IF_MDG_BS_BP_ACCESS_HANDLER`.

If you want to create a custom handler class with inheritance, you can use the following abstract handler classes as parent:

- `CL_MDG_BS_FND_HANDLER` for the foundation layer `MDG_FND`
- `CL_MDG_BS_ECC_HANDLER` for the application layer `MDG_APPL`

Register your custom handler classes for usage in database table `MDG_BS_BP_HNDLR`. Otherwise your classes won't be called by MDG's access class.

## 4. STEP BY STEP EXPLANATION

The following explanation shows you how to create and register a custom handler class.

### 4.1. Create a custom handler class

Start transaction SE24 and create a new class in the customer (Z) namespace.

- Name the new class ZCL\_CUSTOM\_ECC\_HANDLER.
- Maintain the class description as “My custom application handler”.
- Define the class as “Usual ABAP Class”.
- Decide if the class is Final or not.

Switch to the Properties tab and add the CL\_MDG\_BS\_ECC\_HANDLER superclass. With that you can reuse the common implementations and static variables of both class CL\_MDG\_BS\_ECC\_HANDLER and its parent class CL\_MDG\_BS\_FND\_HANDLER.

Save the class. You can also add your custom class to a workbench transport request if you want to use it in other systems according to your system landscape setup.

Since the superclass is abstract, create redefinitions for all methods belonging to the interface IF\_MDG\_BS\_BP\_ACCESS\_HANDLER. The redefinitions themselves can be empty at first. It is enough to implement only those methods that are required to fulfill the needs of your process.

Save, check and activate the class.

### 4.2. Register your custom handler class

Switch to the view maintenance transaction SM30 and select view V\_MDG\_BS\_BP\_HDL.

Create a new entry with Data Model BP and Class/Interface ZCL\_CUSTOM\_ECC\_HANDLER.

Save the new entry. You can also add the registration to a customizing transport request if you want to use it in other systems according to your system landscape setup.

### 4.3. Verify the usage of your custom handler class

The ZCL\_CUSTOM\_ECC\_HANDLER custom handler class is created and registered for the handler processing. To validate its functionality, complete the following steps:

- Set an external breakpoint in the CL\_MDG\_BS\_ECC\_ACCESS class at the end of the PROVIDE\_HANDLER method.
- Start the user interface for customer governance or supplier governance.
- In the debugger, display the GT\_HANDLER\_ACCESS attribute of the CL\_MDG\_BS\_ECC\_ACCESS class.
- Find the ZCL\_CUSTOM\_ECC\_HANDLER custom handler class in the GT\_HANDLER\_ACCESS table.

### 4.4. Add custom logic to your custom handler class

Once your class is defined and registered, add your custom logic in the methods according to your project specific requirements.

#### 4.4.1. Sequence of handler class processing

SAP delivered handler classes will always be processed before your custom class. This behavior is intentionally and will never be changed by SAP.

The sequence ensures that your custom handler can overrule the functionality of the SAP delivered handler classes.

#### **4.4.2. Copying SAP delivered code**

The SAP delivered classes contain a lot of examples how-to structure custom code. You can surely use this code as a template for your own code. But you should never copy the SAP owned code for overwriting or replacing SAP delivered functionality. This is especially important if you redefine methods without a call to the parent class. If you do this, be aware that the copied SAP code cannot be improved by SAP notes anymore. The copied code part of your custom class. SAP cannot provide SAP notes changing the coding of your custom class. If you do this kind of copying, you must check and copy every SAP note code change to your custom class. Otherwise your custom class will run outdated and invalid code, likely causing undesired side-effects.

Your custom handler class shall contain custom code being completely independent from SAP owned code.

## 5. PERFORMANCE

The SAP delivered classes contain a lot of reusable methods and attributes. Familiarize with the SAP owned code and attributes / buffers.

Prevent executing expensive processing logic that was called by the SAP owned classes before. An example is reading active data from the database. The SAP owned handler classes read the complete business partner, including all multiple assignments, customers and suppliers. The data is stored in the database buffers GT\_BP\_DATA\_DB respectively GT\_ECC\_EXTERN\_DB. The buffers contain more tables and fields than supported by the data model BP. If you require data for custom enhancements, check the buffers first. Don't re-read data that is already buffered. Re-read and buffer only those tables and fields that are not covered by the SAP owned code.

SAP uses SMT mappings to map data from the API format into the staging format and vice versa. SMT simplifies custom enhancements. You can easily add and map custom fields for existing SAP tables and entity types. The negative impact is the runtime required by SMT. If you add a custom table to MDG's data model BP, consider coding the mapping in your custom handler class.

Many methods of the handler interface are supplied with a reference of IF\_USMD\_MODEL\_EXT. This reference allows cross reading data that might be not present in the current method but required for its processing logic. You should never use method READ\_ENTITY\_DATA\_ALL. This method is very expensive since it processes all tables and attributes of data model BP. Instead use READ\_CHAR\_VALUE with providing detailed selection criteria in IT\_SEL. This limits the data to be processed and thus ensures a short runtime. Also consider setting parameter IF\_NO\_FLUSH to ABAP\_TRUE to prevent a flush which is very likely not needed by your custom handler class.

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. 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 SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company 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.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.