# How-To Guide: Usage of EAM Custom Extension points for MDG EAM and AIW

## Applies to

MDG EAM Solutions by Prometheus Group

MDG AIW Solutions by Prometheus Group

## Summary

With respect to MDG Standard objects implementation, SAP has provided below mentioned BADI's/Enhancement points to achieve customer specific requirements. For ex for controlling field property control, Derivations, Validations.

1. USMD_ACC_FLD_PROP_CUST_DEP_SET - Access to Customer-Dependent Field Property Settings
2. USMD_RULE_SERVICE_CROSS_ET- Validations/Derivations Across Entity Types
3. USMD_RULE_SERVICE- Define Validations/Derivations

Considering the fact MDG Prometheus products (MDG EAM, MDG AIW) are delivered by using SAP delivered MDG framework, It is common expectation that the above mentioned BADI's/enhancement spots should behave/give results in line with Standard delivered MDG objects.

We have observed that the above mentioned BADI'/Enhancements implementation does not give desired output for MDG EAM /AIW processing due to conflicts within and hence as part of MDG EAM – We have delivered custom extension options to the customer which enables to achieve customer specific projects requirements in replacement to above mentioned BADI/Enhancements.

The Custom extensions points are provided all the major method from MDG EAM access classes which enables customer to write their own custom logic to achieve their customer specific project requirements accordingly. In case if Customer finds this extension, points are missing for any of methods which needs to be customized from customer point of view, please report it to Prometheus group vis OSS – we will validate further for inclusion.

**Author:** Manjunath Goudra

**Company: Prometheus Group**

**Created On: 07-Apr-2025**

**Version: 1.0**

# Table of Contents

# Introduction

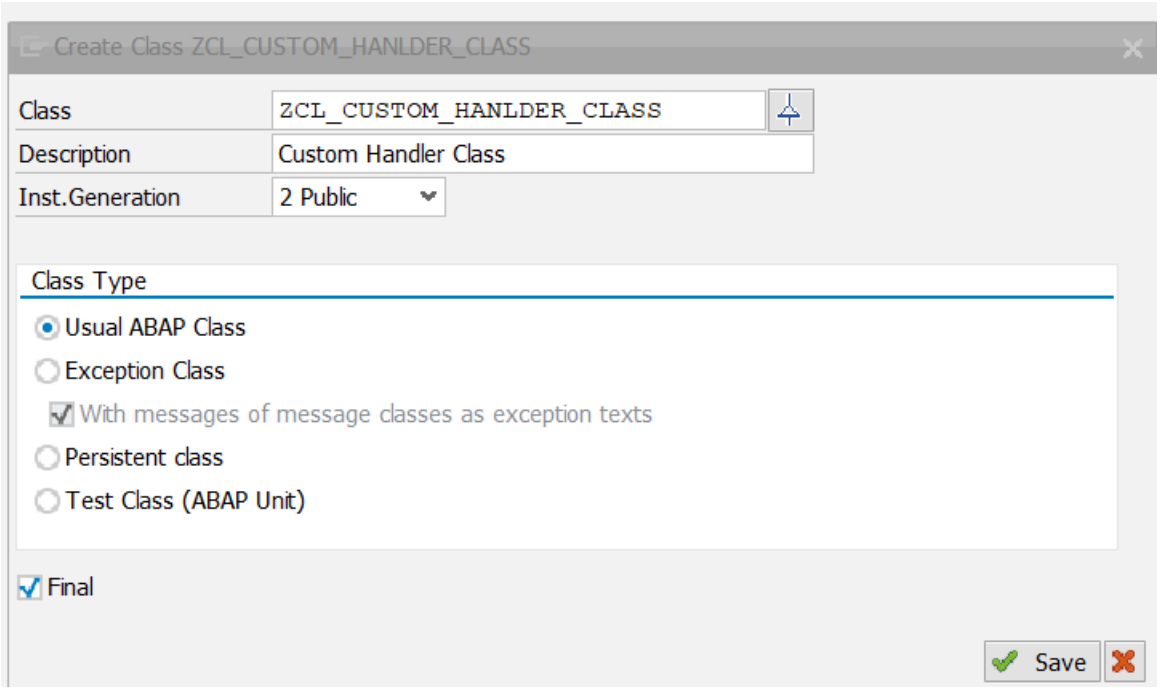
This Guide describes procedure for using the customer extension point delivered in MDG EAM classes to achieve the customer specific project requirements.

# Manual Implementation Steps

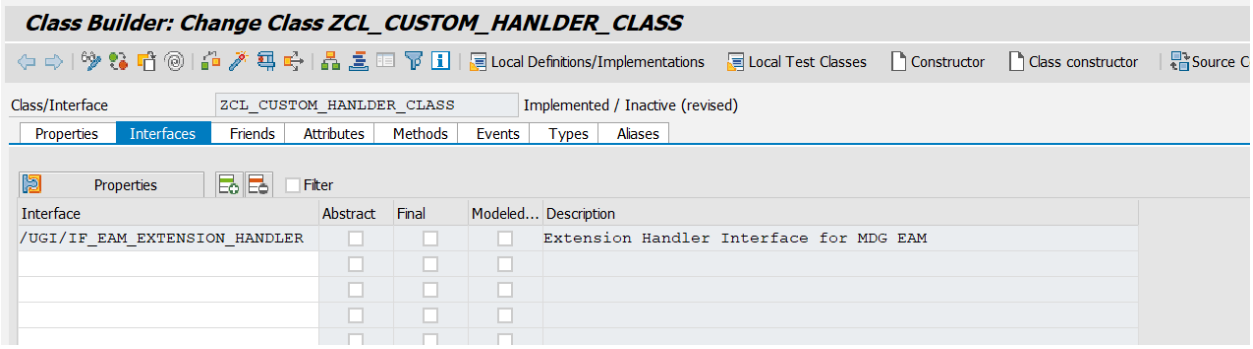
## A. Create a Custom class

Use the below steps :

1. Run t-code SE24 and populate the class name as “ZCL_CUSTOM_HANLDER_CLASS” [ Please create custom class name as per your project guidelines ]




2. Populate the package name and save the changes to relevant TR accordingly as per the your project guidelines.
3. Click on Interface tab and populate the Interface name as “/UGI/IF_EAM_EXTENSION_HANDLER”

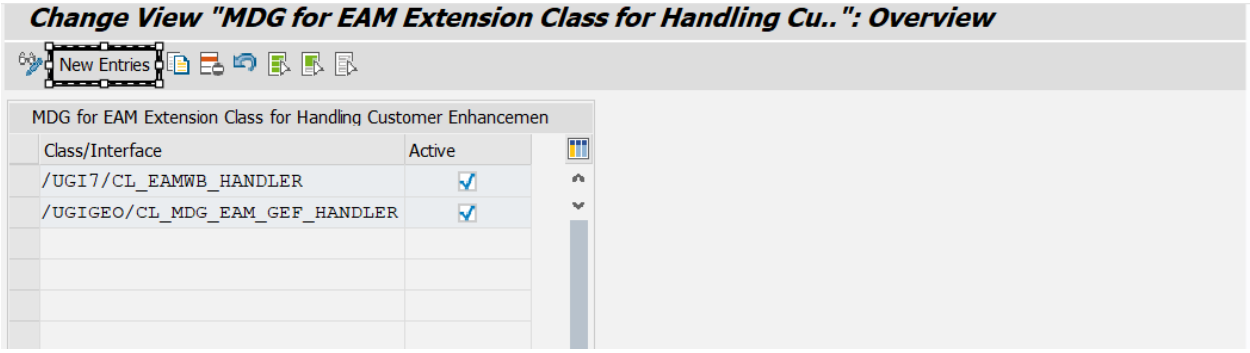



4. Check and save, Activate the class “ZCL_CUSTOM_HANLDER_CLASS”

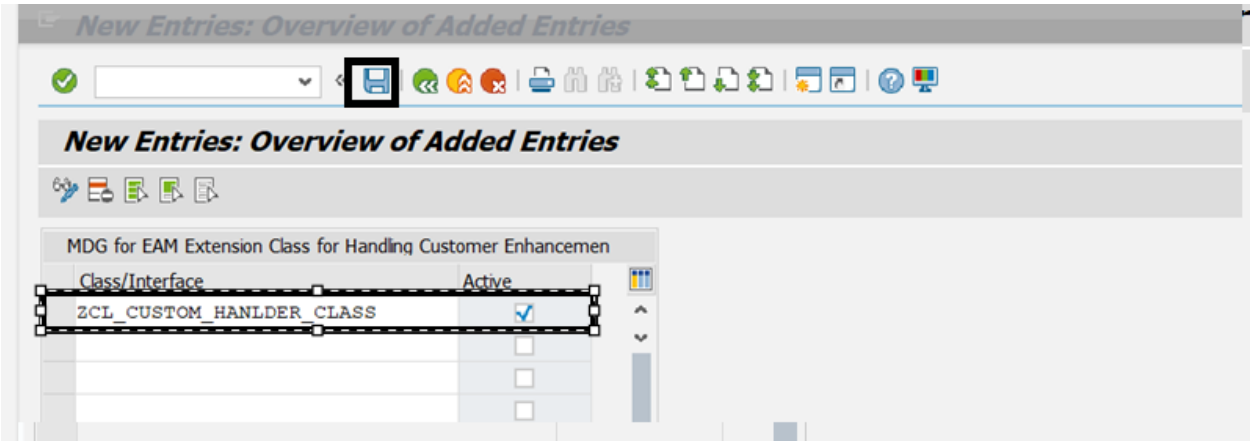
## B. Updates to Table/View “/UGI/MDG_U1_EXT”

Use the below steps

1. Run t-code SM30 and populate the Table/View as “/UGI/MDG_U1_EXT” and Click on Edit and click on button “New Entries”.




2. Populate the class/Interface name as “Class name that you had created in Point A For ex “ZCL_CUSTOM_HANLDER_CLASS” and choose the “Activate” checkbox , Save the details to relevant Transport Request(TR) accordingly .



# C. Writing logic in Custom Class/Method.

1. Run t-code SE24 and Populate the class name as “ZCL_CUSTOM_HANLDER_CLASS” and Click edit.
2. Double click on the method name to have your custom logic and activate the class/method.
3. Please note that relevant method will be called with your custom logic accordingly.

For ex ;

if you want derive a value to given attribute, use the method /UGI/IF_EAM_EXTENSION_HANDLER~DERIVE_DATA to write the custom logic .

If you want to handle field property for given attribute, use the method /UGI/IF_EAM_EXTENSION_HANDLER~GET_FIELD_PROPERTIES

If you want to write any custom logic in searching records , Use the method

/UGI/IF_EAM_EXTENSION_HANDLER~QUERY