



PUBLIC

How to Create a Custom Data Model for a WBS

Applicable Releases:

From MDG 7.0 and from SAP S/4HANA 1511

Version 1.2

October 2024

Document History

Document Version	Description
1.0	First official release of this guide (before 2021)
1.1	Additional information
1.2	Minor changes

Table of contents

1. Introduction.....	5
2. Prerequisites.....	5
3. Background Information.....	6
3.1. Project System.....	6
3.2. MDG Framework.....	8
4. Create a Custom Data Model.....	10
4.1. Concepts and Prerequisites.....	10
4.1.1. Conceptual Data Model.....	10
4.1.2. Logical Data Model.....	10
4.1.3. Physical Data Modelling.....	11
5. Step by Step Explanation.....	12
5.1. Create Business Object Type Code (OTC).....	12
5.2. Creating Entities	12
5.3. Creating Business Objects.....	20
5.3.1. Establishing the Relationships.....	21
5.4. Adding Entities to the WBS Element Entity.....	24
5.5. Representing the Work Breakdown Structure Hierarchically.....	25
5.5.1. Hierarchy Leading Entity	25
5.6. Creating a Custom User Interface.....	26
5.6.1. User Interface Framework.....	27
5.6.2. Develop the Search User Interface.....	28
5.7. Develop a Single Object Maintenance User Interface	32
5.7.1. Configure the Technical UIBB.....	33
5.7.2. Configure Form UIBB for Project Definition.....	37
5.7.3. Configure the List UIBB for Language-Dependent Texts.....	40
5.7.4. Configure the List UIBB for Creating and Changing WBS elements.....	42
5.8. Process Modelling.....	44
5.8.1. Create a New Business Activity.....	45
5.8.2. Assign Business Activities and Logical Actions to Business Objects.....	45
5.8.3. Create Change Request Type	47
5.8.4. Create a Workflow.....	49
5.8.5. Creating Our CR Wire.....	51
5.8.6. Creating Our Role.....	52
6. Testing our Data Model	55
7. Additional Information	57
7.1. Further Reading.....	57
7.1.1. Information on SAP MDG on SAP S/4HANA.....	57
7.1.2. SAP Roadmap Explorer	57
7.1.3. Related Information.....	57

7.2. SAP Notes	57
----------------------	----

1. Introduction

SAP Master Data Governance (MDG) provides business processes to find, create, change, and mark master data for deletion. It supports the governance of 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. MDG offers change request (CR)-based processing of master data with integrated workflow, staging, approval, activation, and distribution.

The SAP MDG custom object framework supports users to model and build MDG applications for the master data objects specific to their business. It can be used also for SAP master data objects for which MDG has not provided any standard data model and applications.

This How To Guide describes the solution to create a custom flex data model for work breakdown structures. The reuse mode, which can be also used if the tables already exists in SAP S/4HANA, is not used in the guide. For simplicity reasons, the guide only describes using flex mode and does not include how to push the staging data to the backend tables.

2. Prerequisites

For this How To Guide you should have access to your system with appropriate access rights.

You will need some basic understanding of the following topics:

- SAP MDG
- SAP Business Workflow
- Floor Plan Manager (FPM)

Development skills are not required as you will not need to write any code during this implementation. If you want to save the configuration into an ABAP transport request, you need a workbench and customizing request/tasks.

3. Background Information

3.1. Project System

Project System (PS) is a project management tool that provides users with support in all phases of their enterprise project. In SAP S/4HANA, PS provides structures that can be used to model and organize project flexibly. PS provides two structures for mapping an enterprise project:

- Work breakdown structures (WBS)
- Networks

A WBS is a model of the project that shows the project deliverables in hierarchical form. WBS' are used to organize a project in the form of a hierarchy and to map the structure of a project. WBS' are made up of WBS elements that are structures in various levels to produce a hierarchy model of the project activities to be carried out.

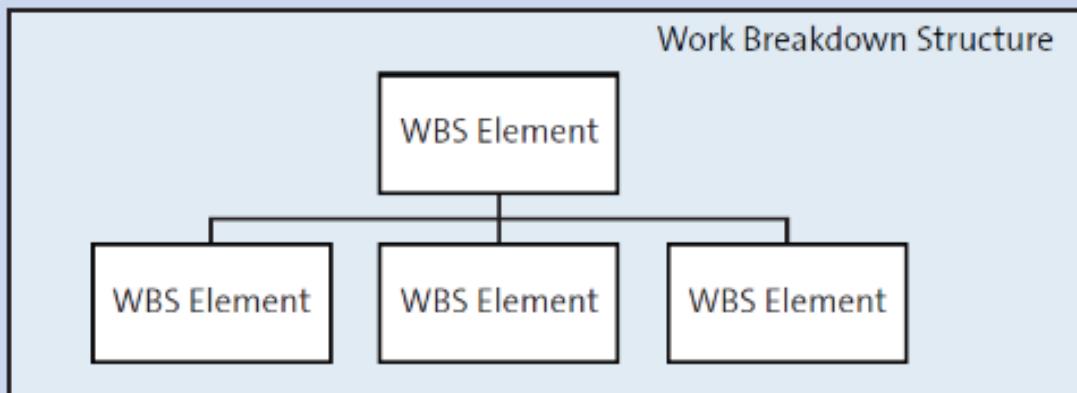
A network represents the course of a project by describing the time sequence and dependencies of events and activities in a project. Networks are used to represent project activities and logical relationships between them.

The following elements are considered master data for the PS application:

- WBS elements
- Networks
- Activities

Depending on the requirements, you can use WBSs, networks, or both to map your project in the SAP system. However, the scope of this guide only includes project definition, WBSs, and WBS elements, not networks and activities.

Each WBS can contain multiple WBS elements as shown in the image below:



The following are different components of WBS:

WBS Elements

WBS elements represent a work package in an enterprise project. WBS elements are actual elements that are used as account assignment objects to record costs, and they can also be used as planning elements. WBS elements are arranged in a hierarchical manner, allowing the data to be summarized at any level.

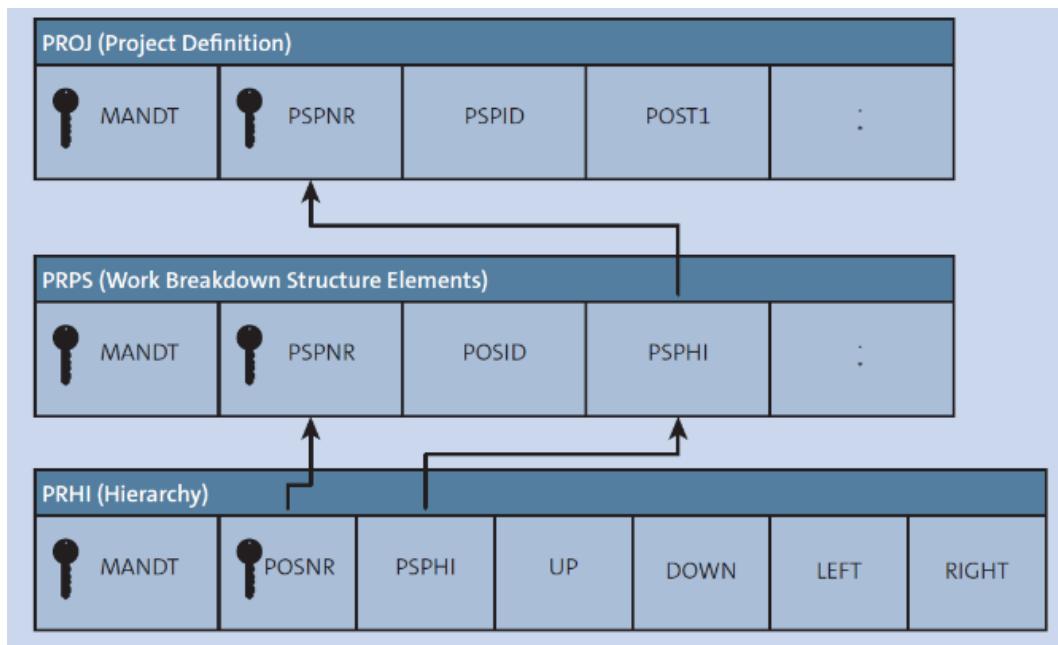
Project Definition

The project definition is used to define the common attributes that are shared across the structures (WBS/network) and structure elements (WBS elements/activities) assigned to the project. Project definition is a mandatory component for creating a project with a network, WBSs, or both. The project definition holds the data that affects the whole project. For example, a controlling area entered in the project is applicable for the whole project. Project definition is also used to define organizational data such as company code, business area, profit centre, and plant. This organizational data is defaulted across the WBS elements.

WBS

The WBS is the model of the project and shows the work packages in a hierarchical structure. Each work package in an enterprise project is represented by WBS elements.

Below are the various tables that together form the data model for PS. All data is saved in the tables. This database architecture will be the basis for the SAP Master Data Governance data model.



3.2. MDG Framework

The MDG foundation framework uses the SAP MDG data model entities, attributes of entities, and relationships between entities to generate the staging area. The staging area is an exclusive persistence layer for MDG, generated from an active SAP MDG data model. The staging area is used to store both active data and inactive data. The goal of the data model is to generate these staging area tables correctly and be the single source of information for relationships between various SAP Master Data Governance entities. The MDG data model is also a source of metadata required for UI modelling. There are two storage modes for active data:

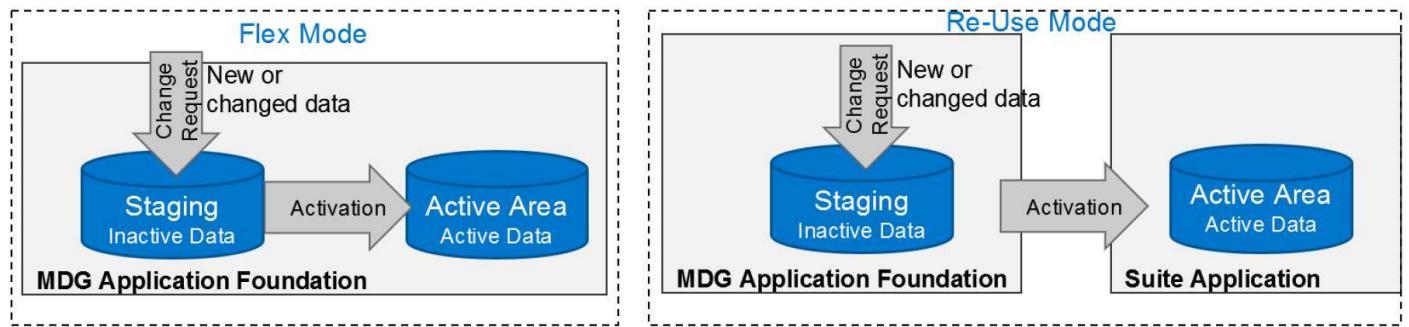
Reuse Mode

This mode is used if the tables needed already exists in SAP S/4HANA. Usually these are the master data objects that are available as part of the SAP S/4HANA data model but aren't delivered as out-of-the-box SAP MDG data models. To use the reuse active area for the custom data models, you must create an active area and assign the access class to the active area.

Flex Mode

This mode is used if no tables are available in SAP S/4HANA. Usually these are the master data objects that aren't available as part of the standard SAP S/4HANA data model. Ideally, the flex option is preferred for business objects that require edition management. SAP S/4HANA doesn't have the edition concept. Instead, these layers enforce time dependency by using valid-from date, valid-to date, or both as key fields in the table.

In our example of WBSs in PS, we'll choose flex mode because the flex model doesn't require the creation of an active area access class, and our current data model involves a hierarchy.



4. Create a Custom Data Model

The first step toward building custom SAP MDG applications is to create a custom data model. The generic interaction layer (genIL) provides uniform API services to access and manipulate underlying business data. The business object layer (BOL) consumes the genIL API. The following sections explain how to create the custom data model, entities, and relationships in detail.

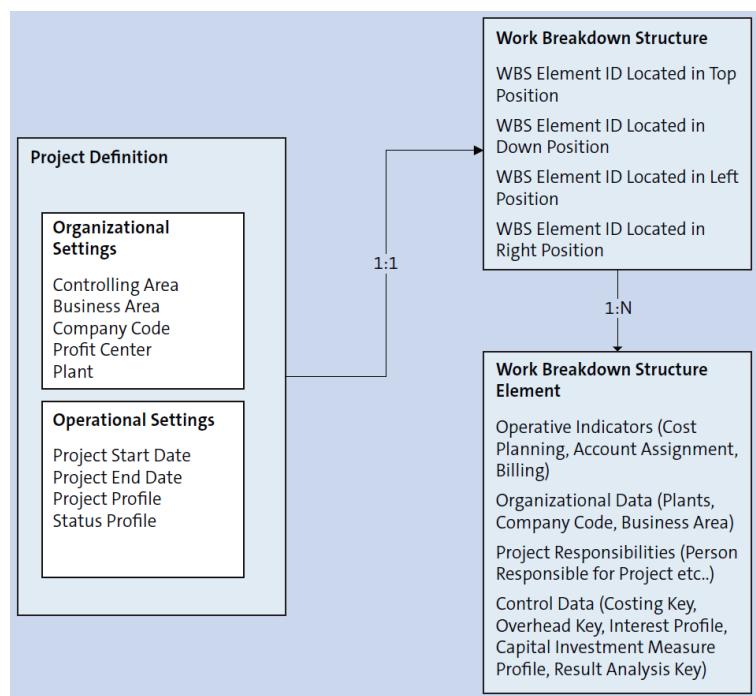
4.1. Concepts and Prerequisites

The process of creating entities and attributes for our custom data model ZX will be described via the following:

- Conceptual data model
- Logical data model
- Physical data model

4.1.1. Conceptual Data Model

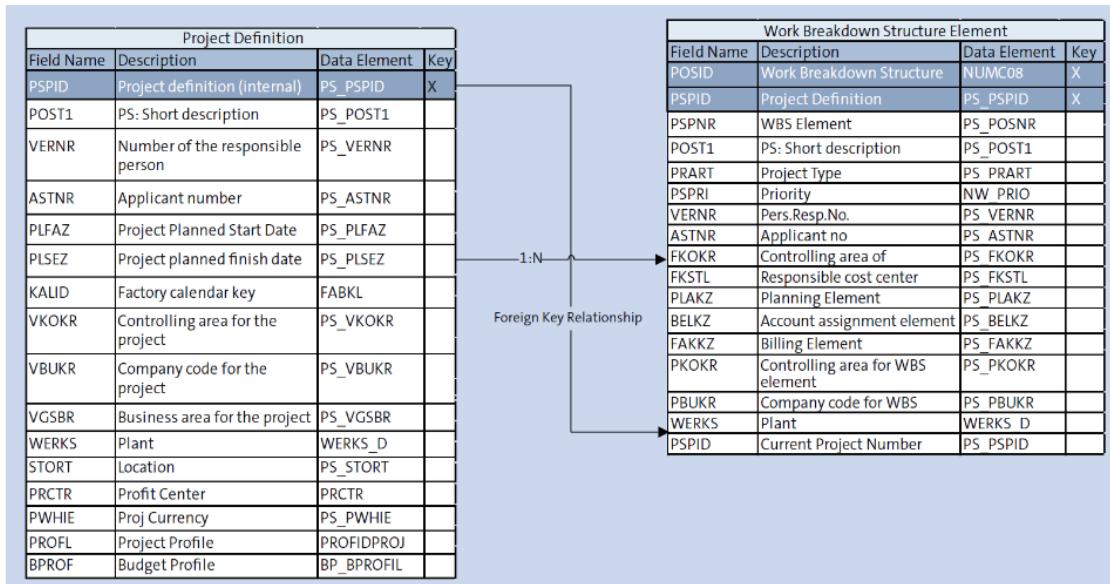
This data model is a high-level representation of the data model architecture. The conceptual model is created with non-technical names make it easy to understand and process for project stakeholders such as executives, business users, and business subject matter experts (SMEs). The conceptual data model acts as the basis for creating the logical data model.



4.1.2. Logical Data Model

The logical data model is a more technical representation of the data model. It displays entities attributes and relationships. The logical data model is normalized by specifying the field-level details such as data type and data length. Below is the logical data model for our custom data model; you can see project definition and

WBS elements along with detailed attributes. The target audience for the logical data model is business SMEs, expert modelers, and application experts.

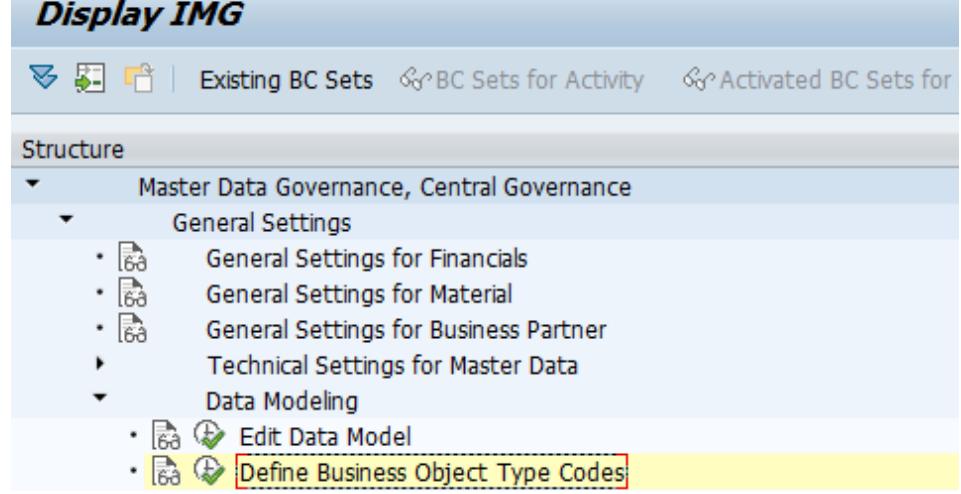
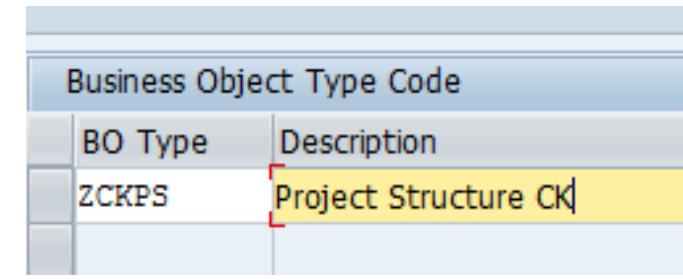


4.1.3. Physical Data Modelling

Finally, the physical data modelling is the process of creating the actual data model in the system. During this step, you take the outcomes (for example, design document) from the preceding two steps and implement them in the system. The physical data model is system specific and deeply technical in nature.

5. Step by Step Explanation

5.1. Create Business Object Type Code (OTC)

<p>Logon with SAP GUI and start transaction MDGIMG.</p> <p>Navigate to <i>General Settings</i> → <i>Data Modeling</i> → <i>Define Business Object Type Codes</i>.</p>	 <p>The screenshot shows the SAP Display IMG interface. In the top menu bar, there are icons for creating new entries (green plus), deleting (red minus), and saving (blue floppy disk). Below the menu, there are three buttons: "Existing BC Sets", "BC Sets for Activity", and "Activated BC Sets for". The main area is titled "Structure" and contains a tree view. The "Master Data Governance, Central Governance" node has a "General Settings" child node. This node has several children: "General Settings for Financials", "General Settings for Material", "General Settings for Business Partner", "Technical Settings for Master Data", "Data Modeling", "Edit Data Model", and "Define Business Object Type Codes". The "Define Business Object Type Codes" node is highlighted with a red dashed border.</p>
<p>Choose <i>New Entries</i> in the top right corner to create your <i>BO Type</i>.</p> <p>Following the correct naming conventions such as Z*.</p> <p>For example: ZCKPS Fill in the <i>Description</i>.</p>	 <p>The screenshot shows a table titled "Business Object Type Code". It has two columns: "BO Type" and "Description". A new row is being added, indicated by a cursor in the "BO Type" column. The "BO Type" cell contains "ZCKPS" and the "Description" cell contains "Project Structure CK". The "Description" cell is highlighted with a yellow background and has a red border around it.</p>
Save your changes.	

5.2. Creating Entities

In the following sections, we'll look at creating entities for your custom data model, starting with type 1 entities before moving on to type 3.

Every data model should have at least one type 1 entity. In the current example, the project definition is one of the type 1 entities.

Return to the MDGIMG screen.

Navigate to **General Settings**
→ **Data Modeling** → **Edit Data Model**.

Alternatively, you can use the Configuration workbench

Display IMG

Existing BC Sets BC Sets for Activity Activated BC Sets for Activity

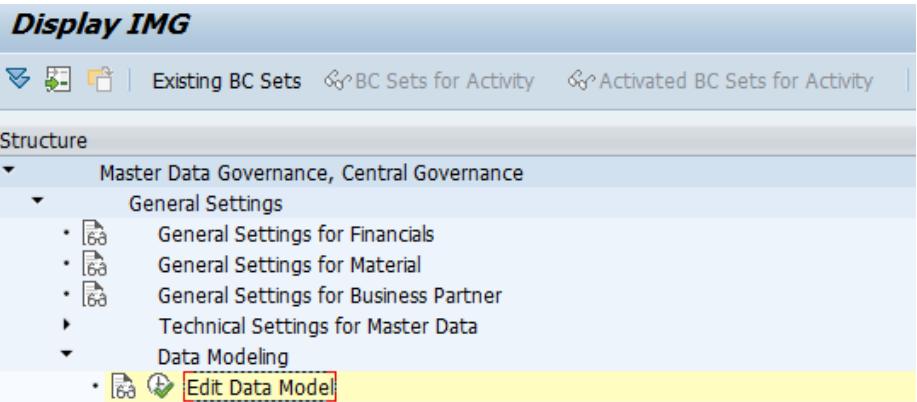
Structure

Master Data Governance, Central Governance

General Settings

- General Settings for Financials
- General Settings for Material
- General Settings for Business Partner
- Technical Settings for Master Data
- Data Modeling

Edit Data Model



Choose New Entries to create a new **Data Model**.

Fill in the details as follows

Data Model: ZX

Description: Data Model for SAP PS - WBS

ActiveArea: provided by MDG

Namespace: ZMDG

Package: ZMDG

Change View "Inactive Data Models": Overview

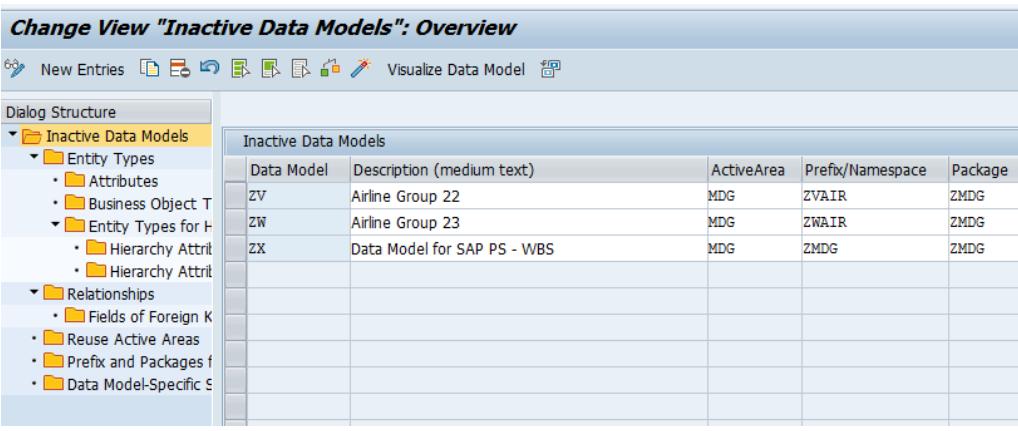
New Entries Visualize Data Model

Dialog Structure

Inactive Data Models

- Entity Types
 - Attributes
 - Business Object T
- Entity Types for H
 - Hierarchy Attrit
 - Hierarchy Attrit
- Relationships
 - Fields of Foreign K
 - Reuse Active Areas
 - Prefix and Packages f
 - Data Model-Specific S

Data Model	Description (medium text)	ActiveArea	Prefix/Namespace	Package
ZV	Airline Group 22	MDG	ZVAIR	ZMDG
ZW	Airline Group 23	MDG	ZWAIR	ZMDG
ZX	Data Model for SAP PS - WBS	MDG	ZMDG	ZMDG



Choose New Entries and add your first Entity Type PSPID.

Storage: Type 1

Data Element: PS_PSPID

1. Edition management has been disabled. In the underlying SAP S/4HANA data model, PS isn't a time dependent entity.
2. Every type 1 entity needs a key assigned.
3. The project definition can have language dependent texts.
4. The Active Area field has been left blank; when left blank the active area of the data model is adopted.

Data Model ZX

Entity Type PSPID

Entity Types

Storage/Use Type	Changeable via Change Request; Generated Database Tables
Validity / Entity	No Edition
Data Element	PS_PSPID
Is Hry Type	No
Validity / Hierarchy	No Edition
Key Assignment	Key Cannot Be Changed; No Internal Key Assignment
<input checked="" type="checkbox"/> Language-Dep. Texts	
Long Text: Length	60
Medium Text: Length	
Short Text: Length	
<input type="checkbox"/> Attachments	
<input type="checkbox"/> Sets	
Search Help	
Src. Fld Short Text	
Src. Fld Medium Text	
Src. Fld Long Text	
Temporary Keys	
Active Area	
Deletion	Deletion Allowed
Description	Project Definition
Structure/Table	
Field	
Struct. X-Flds	

Choose the *Attributes* tab and add new attributes for the domain.

Fill in the attributes as shown.

Data Model ZX

Entity Type PSPID

Attributes

Attribute	Key Field	Data Element	Description
ASTNR	<input type="checkbox"/>	PS_ASTNR	Application Number
KALID	<input type="checkbox"/>	FABKL	Factory Calendar key
PLFAZ	<input type="checkbox"/>	PS_PLFAZ	Projected Planned Start Date
PLSEZ	<input type="checkbox"/>	PS_PLSEZ	Projected planned finish Date
POST1	<input type="checkbox"/>	PS_POST1	PS: Short Description
PROFL	<input type="checkbox"/>	PROFIDPROJ	Project Profile
PWHIE	<input type="checkbox"/>	PS_PWHIE	Proj Currency
VERNR	<input type="checkbox"/>	PS_VERNR	Number Of the Responsible Person
VGSBR	<input type="checkbox"/>	PS_VGSBR	Business Area for the project

Save your changes.

Choose New Entries.

We'll be creating a WBS element as another type 1 entity, but we will assign it the same business object as the project definition (PSPID). By assigning it to the same business object as the project definition, we're ensuring that a WBS element can't be replicated alone, and it needs to be replicated through the project definition.

Fill in the details for the *Entity Types* as shown.

Data Model ZX

Entity Type POSID

Entity Types

Storage/Use Type	Changeable via Change Request; Generated Database Tables
Validity / Entity	No Edition
Data Element	PS_POSID
Is Hry Type	No
Validity / Hierarchy	No Edition
Key Assignment	Key Cannot Be Changed; No Internal Key Assignment
<input checked="" type="checkbox"/> Language-Dep. Texts	
Long Text: Length	60
Medium Text: Length	
Short Text: Length	
<input type="checkbox"/> Attachments	
<input type="checkbox"/> Sets	
Search Help	
Src. Fld Short Text	
Src. Fld Medium Text	
Src. Fld Long Text	
Temporary Keys	
Active Area	
Deletion	Deletion Allowed
Description	WBS Elements

Choose Attributes.

We will be adding an attribute PROJ_DEF. This attribute is used to store the project definition ID of the project to which the WBS element is assigned. This attribute needs to have a foreign key relationship with PSPID.

Data Model ZX

Entity Type POSID

Attributes

Attribute	Key Field	Data Element	Description
PROJ_DEF	<input type="checkbox"/>	PS_PSPID	Project Definition

Save the changes.

The project definition (PSPID) has profit center as one of the attributes. We can't add profit center as an attribute directly under the type 1 entity PSPID because the SAP MDG data model rules dictate that a check table of an entity's attribute can't have more key fields besides the client and key field referring to the attribute.

Entity Type	
PRCTR	
Entity Types	
Storage/Use Type	Not Changeable via MDG; No Generated Tables
Validity / Entity	No Edition
Data Element	PRCTR
Is Hry Type	No
Validity / Hierarchy	No Edition
Key Assignment	Key Cannot Be Changed; No Internal Key Assignment
<input type="checkbox"/> Language-Dep. Texts	
Long Text: Length	<input type="text"/>
Medium Text: Length	<input type="text"/>
Short Text: Length	<input type="text"/>
<input type="checkbox"/> Attachments	
<input type="checkbox"/> Sets	
Search Help	<input type="text"/>
Src. Fld Short Text	<input type="text"/>
Src. Fld Medium Text	<input type="text"/>
Src. Fld Long Text	<input type="text"/>
Temporary Keys	<input type="text"/>
Active Area	<input type="text"/>
Deletion	Deletion Allowed
Description	Profit Centre

The controlling area is present as an attribute both in the project definition and in the WBS element. Therefore, it makes sense to create just one type 3 entity and then assign the same entity as an attribute to the project structure and WBS element entities using referencing relationships.

The controlling area is one of the key fields in the check tables assigned to the profit center therefore, it's imperative that we create the controlling area as a type 2 entity so that the entity can be used to establish a leading relationship.

Entity Type	
Con Area	
Entity Types	
Storage/Use Type	Not Changeable via MDG; No Generated Tables
Validity / Entity	No Edition
Data Element	PS_VKOKR
Is Hry Type	No
Validity / Hierarchy	No Edition
Key Assignment	Key Cannot Be Changed; No Internal Key Assignment
<input type="checkbox"/> Language-Dep. Texts	
Long Text: Length	<input type="text"/>
Medium Text: Length	<input type="text"/>
Short Text: Length	<input type="text"/>
<input type="checkbox"/> Attachments	
<input type="checkbox"/> Sets	
Search Help	<input type="text"/>
Src. Fld Short Text	<input type="text"/>
Src. Fld Medium Text	<input type="text"/>
Src. Fld Long Text	<input type="text"/>
Temporary Keys	<input type="text"/>
Active Area	<input type="text"/>
Deletion	Deletion Allowed
Description	Controlling Area

Like the controlling area the valid to date is a key attribute in the check table so we will create it as entity 3 so we can establish a leading relationship.

Entity Type DATBI	
Entity Types Storage/Use Type: Not Changeable via MDG; No Generated Tables Validity / Entity: No Edition Data Element: DATBI Is Hry Type: No Validity / Hierarchy: No Edition Key Assignment: Key Cannot Be Changed; No Internal Key Assignment	
<input type="checkbox"/> Language-Dep. Texts Long Text: Length <input type="text"/> Medium Text: Length <input type="text"/> Short Text: Length <input type="text"/> <input type="checkbox"/> Attachments <input type="checkbox"/> Sets Search Help <input type="text"/> Src. Fld Short Text <input type="text"/> Src. Fld Medium Text <input type="text"/> Src. Fld Long Text <input type="text"/> Temporary Keys <input type="text"/> Active Area <input type="text"/> Deletion: Deletion Allowed Description: Valid to Date	

Next, we will create the company code as a type 3 entity as an attribute to PSPID using a referencing relationship. We are choosing type 3 as we later want to establish it a leading relationship with the plant entity.

Entity Type BUKRS	
Entity Types Storage/Use Type: Not Changeable via MDG; No Generated Tables Validity / Entity: No Edition Data Element: PS_VBUKR Is Hry Type: No Validity / Hierarchy: No Edition Key Assignment: Key Cannot Be Changed; No Internal Key Assignment	
<input type="checkbox"/> Language-Dep. Texts Long Text: Length <input type="text"/> Medium Text: Length <input type="text"/> Short Text: Length <input type="text"/> <input type="checkbox"/> Attachments <input type="checkbox"/> Sets Search Help <input type="text"/> Src. Fld Short Text <input type="text"/> Src. Fld Medium Text <input type="text"/> Src. Fld Long Text <input type="text"/> Temporary Keys <input type="text"/> Active Area <input type="text"/> Deletion: Deletion Allowed Description: Company code for project	

Plant is used to define organization data during project definition.

To ensure that the SAP Master Data Governance framework understands this relationship between plant and location, we'll create plant as a type 3 entity because we need it to be part of the leading relationship for the location attribute.

Entity Type	WERKS_D
Entity Types	
Storage/Use Type	Changeable w/o Change Request; Generated Check/Text Tables
Validity / Entity	No Edition
Data Element	WERKS_D
Is Hry Type	No
Validity / Hierarchy	No Edition
Key Assignment	Key Cannot Be Changed; No Internal Key Assignment
<input type="checkbox"/> Language-Dep. Texts	
Long Text: Length	<input type="checkbox"/>
Medium Text: Length	<input type="checkbox"/>
Short Text: Length	<input type="checkbox"/>
<input type="checkbox"/> Attachments	
<input type="checkbox"/> Sets	
Search Help	
Src. Fld Short Text	
Src. Fld Medium Text	
Src. Fld Long Text	
Temporary Keys	
Active Area	
Deletion	Deletion Allowed
Description	Plant

The check table assigned to location has plant and location as key fields.

Therefore, as explained before, to be compliant with the SAP MDG data model rules we must:

- Create a location as type 3 entity
- Create a leading relationship between the entity for plant and the entity for location.
- Create a referencing relationship between the entity for location and entity PSPID

Data Model ZX

Entity Type STORT

Entity Types	
Storage/Use Type	Changeable w/o Change Request; Generated Check/Text Tables
Validity / Entity	No Edition
Data Element	PS_STORT
Is Hry Type	No
Validity / Hierarchy	No Edition
Key Assignment	Key Cannot Be Changed; No Internal Key Assignment
<input type="checkbox"/> Language-Dep. Texts	
Long Text: Length	[]
Medium Text: Length	[]
Short Text: Length	[]
<input type="checkbox"/> Attachments	
<input type="checkbox"/> Sets	
Search Help	[]
Src. Fld Short Text	[]
Src. Fld Medium Text	[]
Src. Fld Long Text	[]
Temporary Keys	[]
Active Area	[]
Deletion	Deletion Allowed
Description	Location
Structure/Table	[]
Field	[]
Struct. X-Flds	[]

Save your changes and activate.

5.3. Creating Business Objects

Returning to our list of entities choose the first entity we created, *PSPID*.

Open *Business Object Type* view from the left column.

Data Model ZX			
Entity Types			
Entity Type	Storage/Use Type	Val.Entity	Data Element
BUKRS	Not Changeable via MDG; No ...	No Edition	PS_VBUKR
CON_AREA	Not Changeable via MDG; No ...	No Edition	PS_VKOKR
DATBI	Not Changeable via MDG; No ...	No Edition	DATBI
POSID	Changeable via Change Reque...	No Edition	PS_POSID
PRCTR	Not Changeable via MDG; No ...	No Edition	PRCTR
PSPID	Changeable via Change Reque...	No Edition	PS_PSPID
STORT	Changeable w/o Change Reque...	No Edition	PS_STORT
WERKS_D	Changeable w/o Change Reque...	No Edition	WERKS_D

We can input our business object that we previously created.

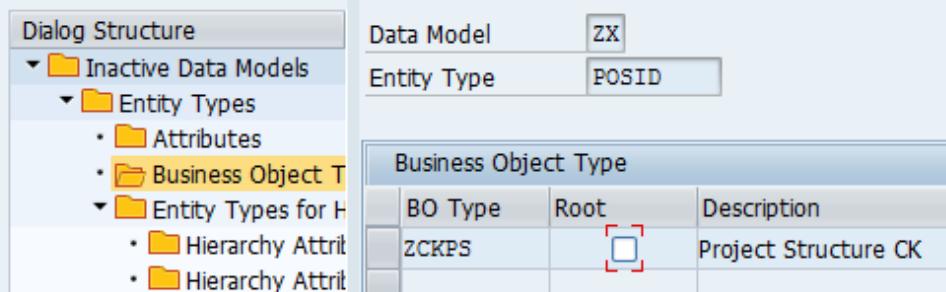
It is important to note that we have selected the *Root* checkbox for this entity.

Change View "Business Object Type": Overview									
Dialog Structure		Data Model ZX							
Inactive Data Models <ul style="list-style-type: none"> Entity Types <ul style="list-style-type: none"> Attributes Business Object T Entity Types for H <ul style="list-style-type: none"> Hierarchy Attrit Hierarchy Attrit 		<input type="text" value="PSPID"/>							
		<table border="1"> <thead> <tr> <th>Business Object Type</th> <th>Root</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ZCKPS</td> <td><input checked="" type="checkbox"/></td> <td>Project Structure CK</td> </tr> </tbody> </table>		Business Object Type	Root	Description	ZCKPS	<input checked="" type="checkbox"/>	Project Structure CK
Business Object Type	Root	Description							
ZCKPS	<input checked="" type="checkbox"/>	Project Structure CK							

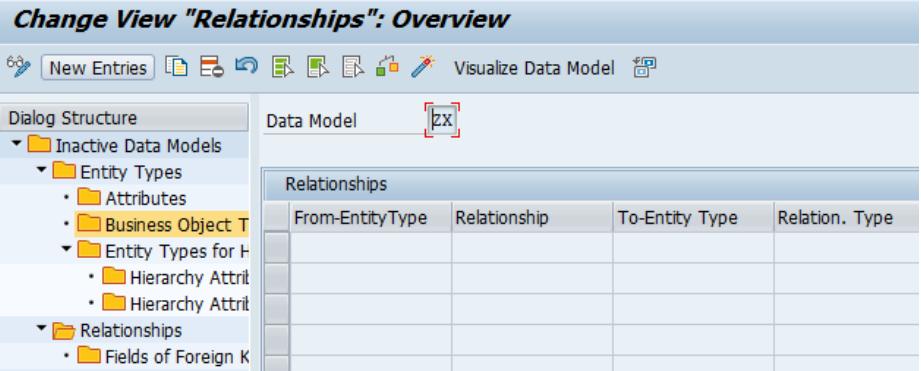
Repeating previous steps, return to our list of entities, this time choosing the 2nd entity we created *POSID*.

Then choose *Business Object Type* from the left column.

Change View "Entity Types": Overview																																							
Dialog Structure		Data Model ZX																																					
Inactive Data Models <ul style="list-style-type: none"> Entity Types <ul style="list-style-type: none"> Attributes Business Object T Entity Types for H <ul style="list-style-type: none"> Hierarchy Attrit Hierarchy Attrit Relationships <ul style="list-style-type: none"> Fields of Foreign K Reuse Active Areas Prefix and Packages f Data Model-Specific S 		<input type="text" value="PSPID"/>																																					
		<table border="1"> <thead> <tr> <th>Entity Types</th> <th>Storage/Use Type</th> <th>Val.Entity</th> <th>Data Element</th> </tr> </thead> <tbody> <tr> <td>BUKRS</td> <td>Not Changeable via MDG; No ...</td> <td>No Edition</td> <td>PS_VBUKR</td> </tr> <tr> <td>CON_AREA</td> <td>Not Changeable via MDG; No ...</td> <td>No Edition</td> <td>PS_VKOKR</td> </tr> <tr> <td>DATBI</td> <td>Not Changeable via MDG; No ...</td> <td>No Edition</td> <td>DATBI</td> </tr> <tr> <td>POSID</td> <td>Changeable via Change Reque...</td> <td>No Edition</td> <td>PS_POSID</td> </tr> <tr> <td>PRCTR</td> <td>Not Changeable via MDG; No ...</td> <td>No Edition</td> <td>PRCTR</td> </tr> <tr> <td>PSPID</td> <td>Changeable via Change Reque...</td> <td>No Edition</td> <td>PS_PSPID</td> </tr> <tr> <td>STORT</td> <td>Changeable w/o Change Reque...</td> <td>No Edition</td> <td>PS_STORT</td> </tr> <tr> <td>WERKS_D</td> <td>Changeable w/o Change Reque...</td> <td>No Edition</td> <td>WERKS_D</td> </tr> </tbody> </table>		Entity Types	Storage/Use Type	Val.Entity	Data Element	BUKRS	Not Changeable via MDG; No ...	No Edition	PS_VBUKR	CON_AREA	Not Changeable via MDG; No ...	No Edition	PS_VKOKR	DATBI	Not Changeable via MDG; No ...	No Edition	DATBI	POSID	Changeable via Change Reque...	No Edition	PS_POSID	PRCTR	Not Changeable via MDG; No ...	No Edition	PRCTR	PSPID	Changeable via Change Reque...	No Edition	PS_PSPID	STORT	Changeable w/o Change Reque...	No Edition	PS_STORT	WERKS_D	Changeable w/o Change Reque...	No Edition	WERKS_D
Entity Types	Storage/Use Type	Val.Entity	Data Element																																				
BUKRS	Not Changeable via MDG; No ...	No Edition	PS_VBUKR																																				
CON_AREA	Not Changeable via MDG; No ...	No Edition	PS_VKOKR																																				
DATBI	Not Changeable via MDG; No ...	No Edition	DATBI																																				
POSID	Changeable via Change Reque...	No Edition	PS_POSID																																				
PRCTR	Not Changeable via MDG; No ...	No Edition	PRCTR																																				
PSPID	Changeable via Change Reque...	No Edition	PS_PSPID																																				
STORT	Changeable w/o Change Reque...	No Edition	PS_STORT																																				
WERKS_D	Changeable w/o Change Reque...	No Edition	WERKS_D																																				

Like before input the business object we previously created.	
In this case leave the Root checkbox deselected.	
Save and activate changes.	

5.3.1. Establishing the Relationships

Choose <i>Relationships</i> from the left column. Then choose <i>New Entries</i> from the tool bar to start creating a relationship.	
As discussed, before we need a foreign key relationship through attribute PROJ_DEF between PSPID and POSID.	
First, we will establish the relationship as shown.	

Then choose *Fields of Foreign Key Relationships*. Here is where we will indicate PROJ_DEF is our foreign key as shown.

Change View "Fields of Foreign Key Relationships": Overview

"From" Field (Check Table Fld)	"To" Field (Foreign Key Field)
PSPID	PROJ_DEF

Next, we will establish the referencing relationships. These are the attributes of project definition we previously created.

Like before return to relationships and choose *New Entries* from the toolbar.

You can enter the relationships as shown.

Data Model ZX

Relationships

From-EntityType	Relations...	To-Entity Type	Relation. Type	Cardinality
BUKRS	VBUKR	PSPID	Referenc...	1 : N
CON_AREA	PKOKR	POSID	Referenc...	1 : N
CON_AREA	VKOKR	PSPID	Referenc...	0 : N
DATBI	DATBI_R	PSPID	Referenc...	0 : N
PRCTR	PRCTR	PSPID	Referenc...	0 : N
STORT	STORT	PSPID	Referenc...	1 : N
WERKS_D	WERKS	PSPID	Referenc...	1 : N

Return to the *Relationships* tab and choose *New Entries* like before.

Here we are going to establish our leading relationships. Both controlling area and valid to date are key attributes within the check table of profit center. Therefore, this is why we are establishing these relationships.

Save and activate.

Data Model ZX

Relationships

From-EntityType	Relations...	To-Entity Type	Relation. Ty...	Cardinality
CON_AREA	KORKS	PRCTR	Leading	1 : N
DATBI	DATBI	PRCTR	Leading	1 : N

We have now almost completed the data model. We have fully created our PSPID entity.

Choose *Visualize Data Model* from the tool bar to view our data model structure.

Change View "Entity Types": Overview																																												
New Entries Visualize Data Model																																												
Dialog Structure		Data Model ZX																																										
Entity Types <ul style="list-style-type: none"> - Attributes - Business Object T - Entity Types for H - Fields of Foreign K - Reuse Active Areas - Prefix and Packages f - Data Model-Specific S 		<table border="1"> <thead> <tr> <th colspan="4">Entity Types</th></tr> <tr> <th>Entity Type</th><th>Storage/Use Type</th><th>Val.Entity</th><th>Data Element</th></tr> </thead> <tbody> <tr> <td>BUKRS</td><td>Not Changeable via MDG; No Ge...</td><td>No Edition</td><td>PS_VBUKR</td></tr> <tr> <td>CON_AREA</td><td>Not Changeable via MDG; No ...</td><td>No Edition</td><td>PS_VKOKR</td></tr> <tr> <td>DATBI</td><td>Not Changeable via MDG; No ...</td><td>No Edition</td><td>DATBI</td></tr> <tr> <td>POSID</td><td>Changeable via Change Reque...</td><td>No Edition</td><td>PS_POSID</td></tr> <tr> <td>PRCTR</td><td>Not Changeable via MDG; No ...</td><td>No Edition</td><td>PRCTR</td></tr> <tr> <td>PSPID</td><td>Changeable via Change Reque...</td><td>No Edition</td><td>PS_PSPID</td></tr> <tr> <td>STORI</td><td>Changeable w/o Change Reque...</td><td>No Edition</td><td>PS_STORI</td></tr> <tr> <td>WERKS_D</td><td>Changeable w/o Change Reque...</td><td>No Edition</td><td>WERKS_D</td></tr> </tbody> </table>			Entity Types				Entity Type	Storage/Use Type	Val.Entity	Data Element	BUKRS	Not Changeable via MDG; No Ge...	No Edition	PS_VBUKR	CON_AREA	Not Changeable via MDG; No ...	No Edition	PS_VKOKR	DATBI	Not Changeable via MDG; No ...	No Edition	DATBI	POSID	Changeable via Change Reque...	No Edition	PS_POSID	PRCTR	Not Changeable via MDG; No ...	No Edition	PRCTR	PSPID	Changeable via Change Reque...	No Edition	PS_PSPID	STORI	Changeable w/o Change Reque...	No Edition	PS_STORI	WERKS_D	Changeable w/o Change Reque...	No Edition	WERKS_D
Entity Types																																												
Entity Type	Storage/Use Type	Val.Entity	Data Element																																									
BUKRS	Not Changeable via MDG; No Ge...	No Edition	PS_VBUKR																																									
CON_AREA	Not Changeable via MDG; No ...	No Edition	PS_VKOKR																																									
DATBI	Not Changeable via MDG; No ...	No Edition	DATBI																																									
POSID	Changeable via Change Reque...	No Edition	PS_POSID																																									
PRCTR	Not Changeable via MDG; No ...	No Edition	PRCTR																																									
PSPID	Changeable via Change Reque...	No Edition	PS_PSPID																																									
STORI	Changeable w/o Change Reque...	No Edition	PS_STORI																																									
WERKS_D	Changeable w/o Change Reque...	No Edition	WERKS_D																																									

Once opened your data model should appear as shown. You can see the difference relationships that we established.

Detail View Active Version					
Data Model	Name	Data Element	Referenced E...
ZX					
PSPID	Project Definition				
PSPID	Project definition E...	<input checked="" type="checkbox"/>	PS_PSPID		
ASTNR	Applicant no.	A...	<input type="checkbox"/>	PS_ASTNR	
DATBI_R	Valid To	A...	<input type="checkbox"/>	DATBI	DATBI
KALID	Factory Calendar	A...	<input type="checkbox"/>	FABKL	
PLFAZ	Start date	A...	<input type="checkbox"/>	PS_PLFAZ	
PLSEZ	Finish date	A...	<input type="checkbox"/>	PS_PLSEZ	
POST1	Description	A...	<input type="checkbox"/>	PS_POST1	
PRCTR	Profit Center	A...	<input type="checkbox"/>	PRCTR	PRCTR
PROFL	Project Profile	A...	<input type="checkbox"/>	PROFIDPROJ	
PWHIE	Project Currency	A...	<input type="checkbox"/>	PS_PWHIE	
STORT	Location	A...	<input type="checkbox"/>	PS_STORT	STORT
TXTLG	Description (lo...	A...	<input type="checkbox"/>	USMD_TXTLG	
VBUKR	Company code	A...	<input type="checkbox"/>	PS_VBUKR	BUKRS
VERNR	No. of Person ...	A...	<input type="checkbox"/>	PS_VERNR	
VGSBR	Business area	A...	<input type="checkbox"/>	PS_VGSBR	
VKOKR	Controlling area	A...	<input type="checkbox"/>	PS_VKOKR	CON_AREA
WERKS	Plant	A...	<input type="checkbox"/>	WERKS_D	WERKS_D

5.4. Adding Entities to the WBS Element Entity

We need to add attributes to our POSID entity.

Returning to our entity list choose *POSID* and choose *Attributes* from the left column as done previously.

The choose *New Entries*.

Among the attributes that need to be added to the *POSID* entity there are simple attributes and modelled attributes. Simple attributes can be added through the attribute tab whereas modelled need to be added through relationships.

Shown is the list of simple attributes, add these as shown.

Change View "Attributes": Overview											
Dialog Structure		Data Model									
<ul style="list-style-type: none"> - Attributes - Business Object Type - Entity Types for Hierarc <ul style="list-style-type: none"> - Hierarchy Attributes - Hierarchy Attributes 		<input type="checkbox"/> ZX Entity Type <input type="text" value="POSID"/>									
Attributes											
<table border="1"> <thead> <tr> <th>Attribute</th> <th>Key Field</th> <th>Data Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>PROJ_DEF</td> <td><input type="checkbox"/></td> <td>PS_PSPID</td> <td></td> </tr> </tbody> </table>				Attribute	Key Field	Data Element	Description	PROJ_DEF	<input type="checkbox"/>	PS_PSPID	
Attribute	Key Field	Data Element	Description								
PROJ_DEF	<input type="checkbox"/>	PS_PSPID									

Attributes			
Attribute	Key Field	Data Element	Description
ASTNR	<input type="checkbox"/>	PS_ASTNR	Applicant no
BELKZ	<input type="checkbox"/>	PS_BELKZ	Account Assignment element
FAKKZ	<input type="checkbox"/>	PS_FAKKZ	Billing Element
PLAKZ	<input type="checkbox"/>	PS_PLAKZ	Planning Element
POST1	<input type="checkbox"/>	PS_POST1	PS:Short Description
PRART	<input type="checkbox"/>	PS_PRART	Project Type
PSPRI	<input type="checkbox"/>	NW_PRIOR	priority
PS_POSID	<input type="checkbox"/>	PS_POSID	Pers.Resp No
VERNR	<input type="checkbox"/>	PS_VERNR	

As before to establish relationships, choose *Relationship* from the left column and choose *New Entries*.

Add Relationships as shown.

Relationships					
From-EntityType	Relationship	To-EntityType	Relation. Type	Cardinality	
BUKRS	PBUKR	POSID	Referencing	1 : N	
CON_AREA	PKOKR	POSID	Referencing	1 : N	
WERKS_D	WERKS_W	POSID	Referencing	1 : N	

Save and activate.

Following the steps in 5.3.1 Establishing the Relationships, choose the *Visualize Data Model* button from the tool bar.

Your data model should appear as shown.

Data Model	Name	E..	i..	Data Element	Referenced
ZX					
PSPID	Project Definition				
POSID	WBS Elements				
POSID	WBS element	E...	<input checked="" type="checkbox"/>	PS_POSID	
ASTNR	Applicant no.	A...	<input type="checkbox"/>	PS_ASTNR	
BELKZ	Acct asst elem.	A...	<input type="checkbox"/>	PS_BELKZ	
FAKKZ	Billing Element	A...	<input type="checkbox"/>	PS_FAKKZ	
PBUKR	Company code	A...	<input type="checkbox"/>	PS_VBUKR	BUKRS
PKOKR	Controlling area	A...	<input type="checkbox"/>	PS_VKOKR	CON_AREA
PLAKZ	Planning Element	A...	<input type="checkbox"/>	PS_PLAKZ	
POST1	Description	A...	<input type="checkbox"/>	PS_POST1	
PRART	Project Type	A...	<input type="checkbox"/>	PS_PRART	
PROJ_DEF	Project definition	A...	<input checked="" type="checkbox"/>	PS_PSPID	PSPID
PSPRI	Priority	A...	<input type="checkbox"/>	NW_PRIO	
PS_POSID	WBS element	A...	<input type="checkbox"/>	PS_POSID	
TXTLG	Description (lo...	A...	<input type="checkbox"/>	USMD_TXTLG	
VERNR	No. of Person ...	A...	<input type="checkbox"/>	PS_VERNR	
WERKS_W	Plant	A...	<input type="checkbox"/>	WERKS_D	WERKS_D

5.5. Representing the Work Breakdown Structure Hierarchically

Before we begin, it's important to emphasize that WBSs and WBS elements are different. A WBS is a hierarchical representation of WBS elements, whereas the WBS element represents a work package in a WBS. Entity PSPID represents the project definition, and entity POSID represents the WBS element and not the WBS.

5.5.1. Hierarchy Leading Entity

The hierarchy leading entity is the type 1 entity in which the hierarchy setting is activated. In our example, we'll configure entity POSID as the leading entity.

Return to our list of entities and choose *POSID*.

Then choose the *Details* button from the toolbar.

Display View "Entity Types": Overview				
Visualize Data Model				
Data Model ZX				
Entity Type	Storage/Use Type	Val.Entity	Data Element	
BUKRS	Not Changeable via MDG; No ...	<input type="checkbox"/>	No Edition	PS_VBUKR
CON_AREA	Not Changeable via MDG; No ...	<input type="checkbox"/>	No Edition	PS_VKOKR
DATBI	Not Changeable via MDG; No ...	<input type="checkbox"/>	No Edition	DATBI
POSID	Changeable via Change Reque...	<input type="checkbox"/>	No Edition	PS_POSID
PRCTR	Not Changeable via MDG; No ...	<input type="checkbox"/>	No Edition	PRCTR

Change the *Is Hry Type* to
Yes – Not Version-
Dependent/ Synchronized

Data Model	ZX
Entity Type	POSID
Entity Types	
Storage/Use Type	Changeable via Change Request; Generated Database Tables
Validity / Entity	No Edition
Data Element	PS_POSID
Is Hry Type	Yes - Not Version-Dependent / Synchronized
Validity / Hierarchy	No Edition
Key Assignment	Key Cannot Be Changed; No Internal Key Assignment
<input checked="" type="checkbox"/> Language-Dep. Texts	

Next, we will return to our entity list and select *POSID*.

This time we will choose
Hierarchy Attributes.

Display View "Entity Types": Overview

Entity Types			
Entity Type	Storage/Use Type	Val.Entity	Data Element
BUKRS	Not Changeable via MDG; No ...	No Edition	PS_VBUKR
CON_AREA	Not Changeable via MDG; No ...	No Edition	PS_VKOKR
DATBI	Not Changeable via MDG; No ...	No Edition	DATBI
POSID	Changeable via Change Reque...	No Edition	PS_POSID
PRCTR	Not Changeable via MDG; No ...	No Edition	PRCTR

Add *PSPID* and choose
Hierarchy Name as entity
Use.

Change View "Entity Types for Hierarchies": Overview

Entity Types for Hierarchies	
Ent.Type of Node	Use
PSPID	Hierarchy Name

5.6. Creating a Custom User Interface

This section introduces the concept of building the MDG application UI using the SAP MDG custom object UI framework.

5.6.1. User Interface Framework

The major building blocks of the custom object UI framework are as follows:

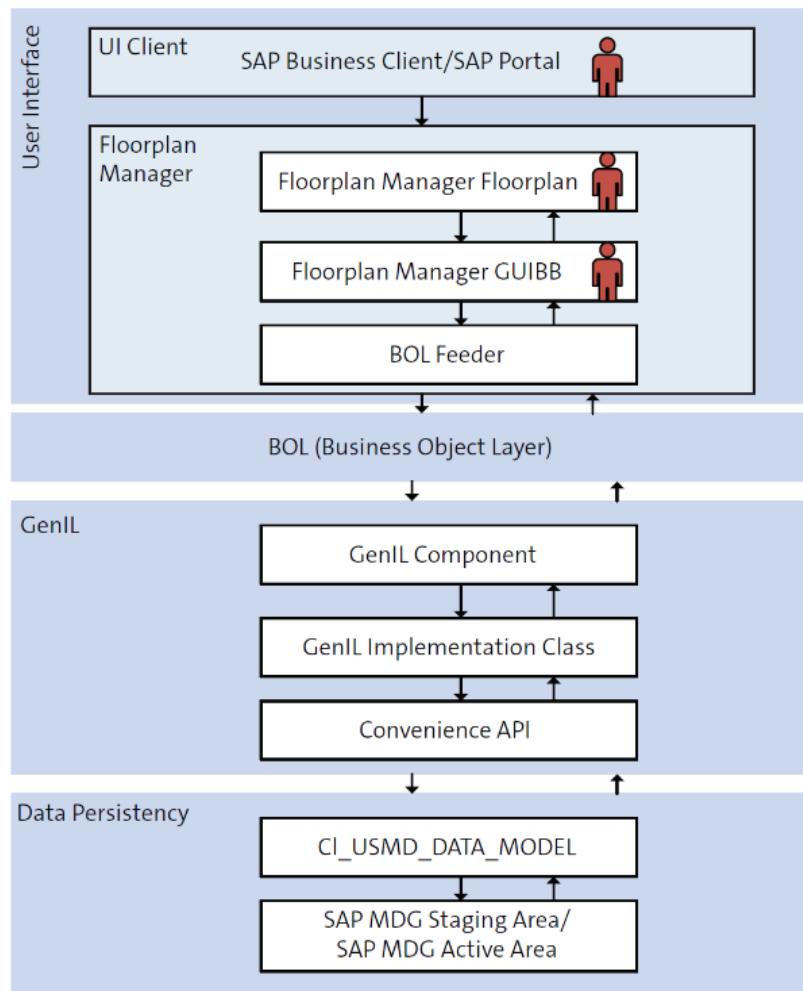
User Interface:

The UI technology for custom object UI is Web Dynpro-based Floorplan Manager. Floorplan Manager enforces consistency in the UI and compliance with the UI guidelines. By providing generic UI building blocks (GUIBB) and predefined floorplans.

USMD_OVP_GEN is the generic Web Dynpro application delivered by SAP to validate the UI for custom object applications.

GenIL and Business Object Layer (BOL):

The purpose of genIL is to provide uniform access to the underlying data persistency layer. It encapsulates business object-specific implementation and provides a uniform interface to access data from the persistency layer.

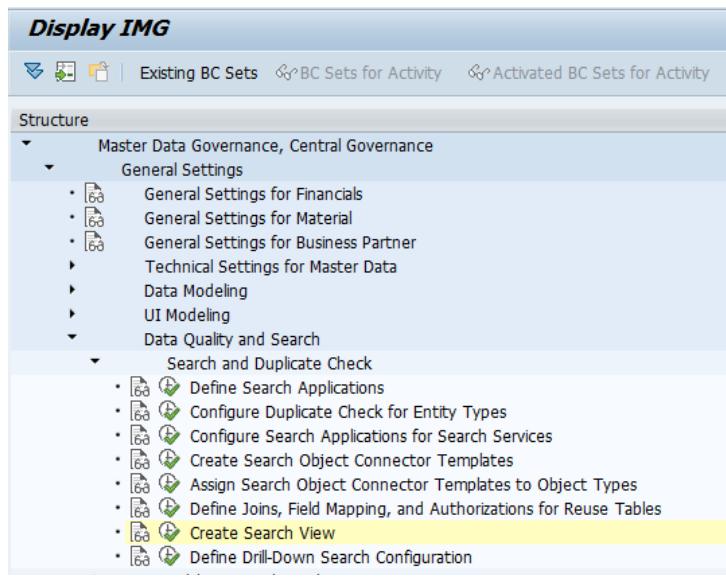
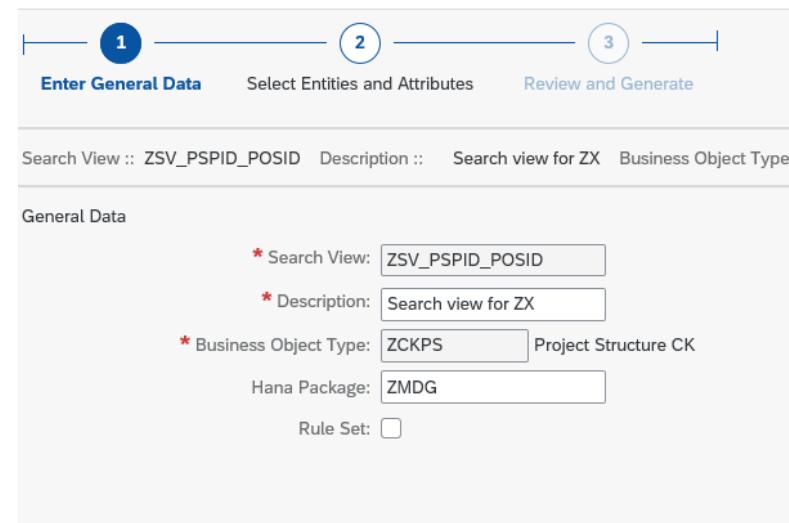
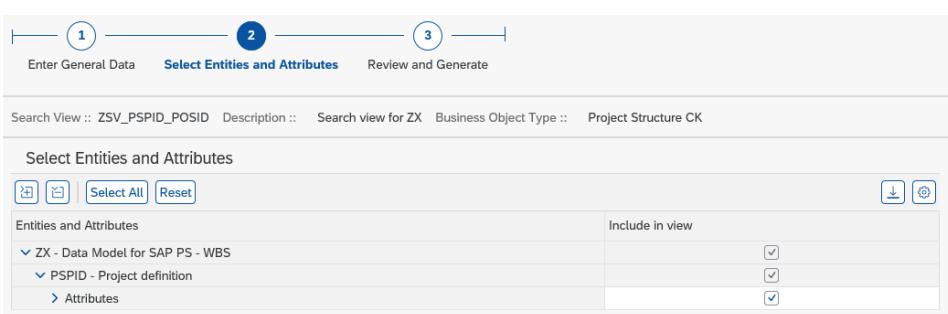


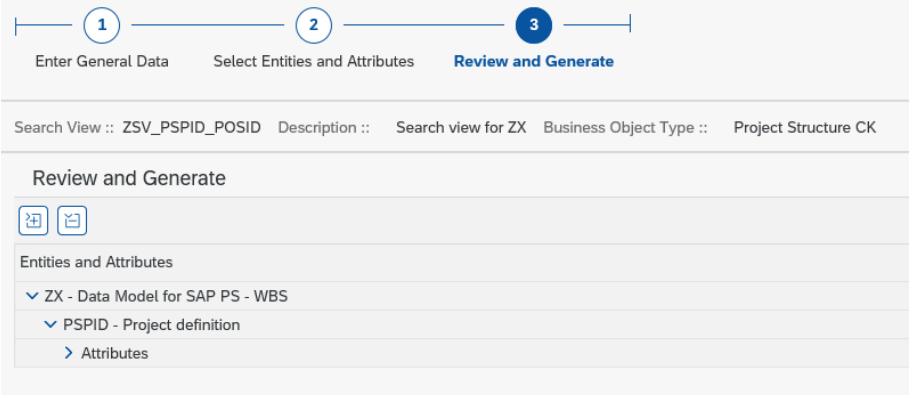
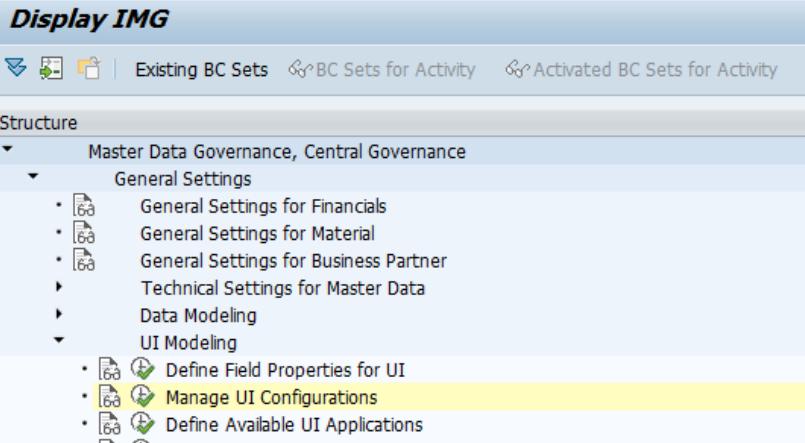
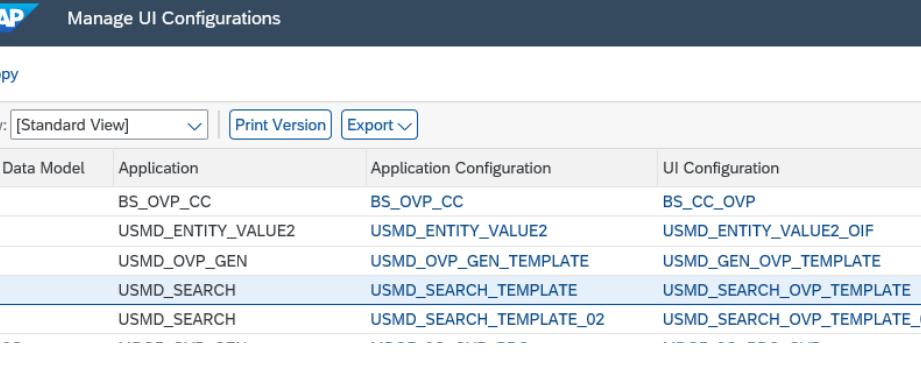
The major building blocks of the custom object UI framework are shown in the image. This is the runtime architecture and building blocks of the custom object UI framework.

The loose coupling of the various building blocks guarantees minimum disruption. For example, if the SAP OData framework has a BOL adapter, then we can easily replace the Floorplan Manager with SAP Fiori without changing the code in the underlying genIL and BOL framework.

5.6.2. Develop the Search User Interface

SAP Master Data Governance provides a generic search application called USMD_SEARCH, which can be used to search any data model and type 1 entity using any search provider.

<p>Our first step toward enabling the generic search UI is to create a search help.</p> <p>Run transaction MDGIMG and navigate to General Settings → Data Quality and Search → Search and Duplicate Check → Create Search View.</p> <p>Then choose New to create a new search help.</p>	
<p>Fill out the general data as shown.</p> <p>Insert a name in the Search View tab. Input the business object we created.</p> <p>Then choose Next.</p>	
<p>Open the PSPID tab and ensure that attributes are selected to be <i>Included in view</i>.</p> <p>Choose Next.</p>	

<p>Check all entities and attributes are included.</p> <p>If so, choose Save.</p> <p>You will now find your search view among the list of MDG search views.</p>																									
<p>Open the <i>Manage UI Configurations</i> Customizing by navigating to General Settings → UI Modeling → Manage UI Configurations.</p> <p>This Customizing activity is used to manage all the object maintenance-related UIs.</p>																									
<p>We are going to create a new generic search application.</p> <p>Select the application configuration USMD_SEARCH_TEMPLATE and choose Copy.</p>	 <table border="1"> <thead> <tr> <th>Data Model</th> <th>Application</th> <th>Application Configuration</th> <th>UI Configuration</th> </tr> </thead> <tbody> <tr> <td>BS_OVP_CC</td> <td>BS_OVP_CC</td> <td>BS_OVP_CC</td> <td>BS_CC_OVP</td> </tr> <tr> <td>USMD_ENTITY_VALUE2</td> <td>USMD_ENTITY_VALUE2</td> <td>USMD_ENTITY_VALUE2</td> <td>USMD_ENTITY_VALUE2_OIF</td> </tr> <tr> <td>USMD_OVP_GEN</td> <td>USMD_OVP_GEN</td> <td>USMD_OVP_GEN_TEMPLATE</td> <td>USMD_GEN_OVP_TEMPLATE</td> </tr> <tr style="background-color: #ADD8E6;"> <td>USMD_SEARCH</td> <td>USMD_SEARCH</td> <td>USMD_SEARCH_TEMPLATE</td> <td>USMD_SEARCH_OVP_TEMPLATE</td> </tr> <tr> <td>USMD_SEARCH</td> <td>USMD_SEARCH</td> <td>USMD_SEARCH_TEMPLATE_02</td> <td>USMD_SEARCH_OVP_TEMPLATE_02</td> </tr> </tbody> </table>	Data Model	Application	Application Configuration	UI Configuration	BS_OVP_CC	BS_OVP_CC	BS_OVP_CC	BS_CC_OVP	USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2_OIF	USMD_OVP_GEN	USMD_OVP_GEN	USMD_OVP_GEN_TEMPLATE	USMD_GEN_OVP_TEMPLATE	USMD_SEARCH	USMD_SEARCH	USMD_SEARCH_TEMPLATE	USMD_SEARCH_OVP_TEMPLATE	USMD_SEARCH	USMD_SEARCH	USMD_SEARCH_TEMPLATE_02	USMD_SEARCH_OVP_TEMPLATE_02
Data Model	Application	Application Configuration	UI Configuration																						
BS_OVP_CC	BS_OVP_CC	BS_OVP_CC	BS_CC_OVP																						
USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2_OIF																						
USMD_OVP_GEN	USMD_OVP_GEN	USMD_OVP_GEN_TEMPLATE	USMD_GEN_OVP_TEMPLATE																						
USMD_SEARCH	USMD_SEARCH	USMD_SEARCH_TEMPLATE	USMD_SEARCH_OVP_TEMPLATE																						
USMD_SEARCH	USMD_SEARCH	USMD_SEARCH_TEMPLATE_02	USMD_SEARCH_OVP_TEMPLATE_02																						

First thing we will do is change the affixes. To do this choose *Change Affixes*, and we will input our data model here **ZX**.

Choose **OK**.

Change Affixes of Target Configuration IDs

Prefix:

Suffix:

OK **Cancel**

Next, we can rename the *Target Configuration ID*, by removing the template.

We can also deselect the *Copy* checkbox for *Overview Page Floorplan*.

Choose *Deep-Copy Mode*.

SAP Floorplan Manager: Application Hierarchy Browser - Application: USMD_SEARCH

Browser Mode Deep-Copy Mode

Application Hierarchy: Configuration Level

Hierarchy	Copy	Component	Interface...	Configuration ID	Target Configuration ID
Application Configuration	<input checked="" type="checkbox"/>			USMD_SEARCH_TEMPLATE	ZX_USMD_SEARCH
Overview Page Floorplan	<input type="checkbox"/>	FPM_OVP_COM...		USMD_SEARCH_OVP_TEMPLATE	
Search					
SECTION_1					
FPM_SEARCH_UIBB	<input type="checkbox"/>	FPM_SEARCH....	SEARCH...	USMD_SEARCH_DQUERY_TEMP...	
FPM_LIST_UIBB_ATS	<input type="checkbox"/>	FPM_LIST_UIBB...	LIST_WI...	USMD_SEARCH_RESULT_TEMP...	

Return to the list of UI configurations, find the UI configuration we just created.

Then choose the *Application Configuration* link **ZX_USMD_SEARCH**.

SAP Manage UI Configurations

Copy

View: [Standard View]  

Data Model	Application	Application Configuration	UI Configuration
BS_OVP_CC	BS_OVP_CC	BS_OVP_CC	BS_CC_OVP
USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2_OIF
USMD_OVP_GEN	USMD_OVP_GEN	USMD_OVP_GEN_TEMPLATE	USMD_GEN_OVP_TEMPLATE
USMD_SEARCH	USMD_SEARCH	USMD_SEARCH_TEMPLATE	USMD_SEARCH_OVP_TEMPLATE
USMD_SEARCH	USMD_SEARCH	USMD_SEARCH_TEMPLATE_02	USMD_SEARCH_OVP_TEMPLATE_02
ZX_USMD_SEARCH	ZX_USMD_SEARCH	ZX_USMD_SEARCH	USMD_SEARCH_OVP_TEMPLATE
OG	MDGF_OGP_OVP_GEN	MDGF_OGP_OVP_BDC	MDGF_OGP_BDC_OVP
OG	MDGF_OVP_GEN	MDGF_OVP_BDCSET	MDGF_OGP_BDCSET_OVP

<p>Choose the pencil icon in the top left corner to edit.</p> <p>Then fill in the following values.</p>	<p>USMD_HIERARCHY): <input type="text"/></p> <p>(USMD_MODEL): <input type="text"/></p> <p>(USMD_OTC): <input type="text"/> ZCKPS</p> <p>(USMD_PROCESS): <input type="text"/></p> <p>CH_EDITION_MODE): <input type="text"/></p> <p>MD_SEARCH_HELP): <input type="text"/> ZSV_PSPID_POSID</p> <p>MD_SEARCH_MODE): <input type="text"/> HA</p>
Choose Save and exit.	

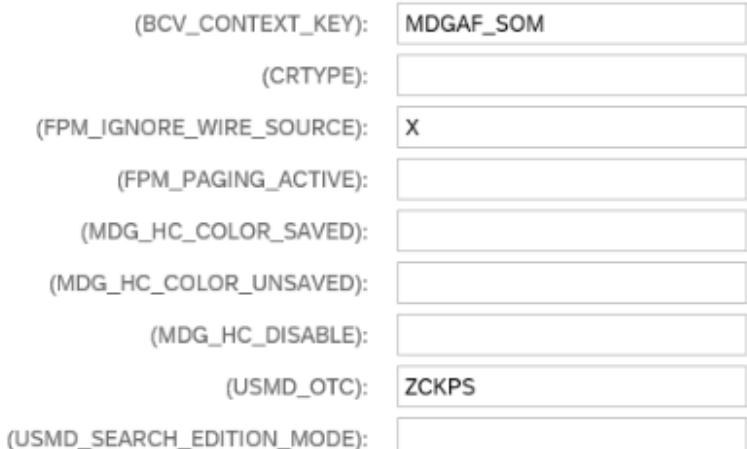
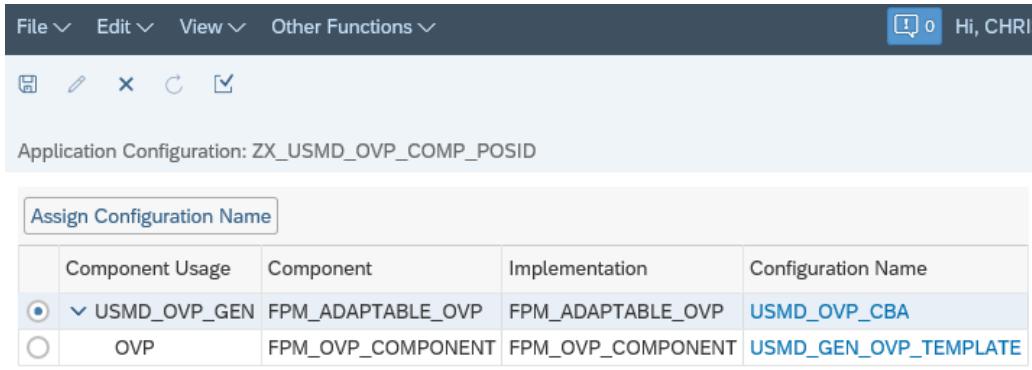
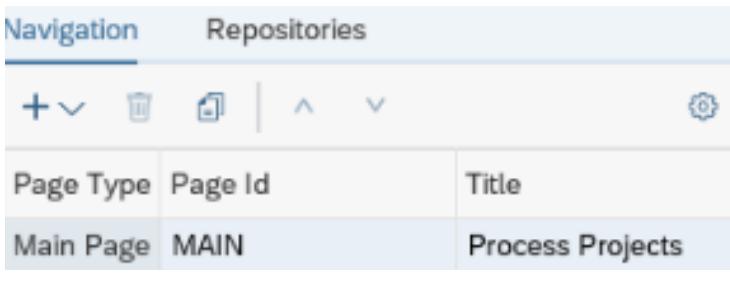
One very important step that needs to be performed after copying the generic search template configuration is to create the communicator (MDG_BS_GOV_COMMUNICATOR) configuration. The name of the communicator configuration and the search UI application must be the same. If they aren't, then the search UI application won't be rendered correctly. The communicator is responsible for rendering the search criteria and search results areas based on the search help configuration.

<p>Now that we made changes you can see our UI configuration has our data model assigned to it.</p> <p>Choose Details.</p>	<table border="1"> <tr> <td><input type="radio"/></td><td>Y0</td><td>USMD_OVP_GEN</td><td>Y0_OVP_COUNTRY</td></tr> <tr> <td><input type="radio"/></td><td>Y0</td><td>USMD_OVP_GEN</td><td>Y0_OVP_COUNTRY_RS</td></tr> <tr> <td><input type="radio"/></td><td>Z0</td><td>USMD_ENTITY_VALUE2</td><td>ZZ0_USMD_ENTITY_VALUE2</td></tr> <tr> <td><input type="radio"/></td><td>ZX</td><td>USMD_SEARCH</td><td>ZX_USMD_SEARCH</td></tr> </table>	<input type="radio"/>	Y0	USMD_OVP_GEN	Y0_OVP_COUNTRY	<input type="radio"/>	Y0	USMD_OVP_GEN	Y0_OVP_COUNTRY_RS	<input type="radio"/>	Z0	USMD_ENTITY_VALUE2	ZZ0_USMD_ENTITY_VALUE2	<input type="radio"/>	ZX	USMD_SEARCH	ZX_USMD_SEARCH
<input type="radio"/>	Y0	USMD_OVP_GEN	Y0_OVP_COUNTRY														
<input type="radio"/>	Y0	USMD_OVP_GEN	Y0_OVP_COUNTRY_RS														
<input type="radio"/>	Z0	USMD_ENTITY_VALUE2	ZZ0_USMD_ENTITY_VALUE2														
<input type="radio"/>	ZX	USMD_SEARCH	ZX_USMD_SEARCH														
<p>Simply ensure that your component and configuration is set as is.</p>	 <p>The screenshot shows the SAP Fiori interface for 'Component Configuration'. At the top, there's a toolbar with 'Delete', 'Copy', 'Other Functions', and a user icon ('Hi, CHRIS!'). Below the toolbar, the title 'Component Configuration' is displayed. The configuration details are listed as follows:</p> <ul style="list-style-type: none"> Component Name: <input type="text"/> MDG_BS_GOV_COMMUNICATOR * Configuration ID: <input type="text"/> ZX_USMD_SEARCH <p>At the bottom of the configuration section, there are three buttons: 'Change' (highlighted in blue), 'Display', and 'Create'.</p>																
<p>On the next screen, change nothing, and choose Save.</p>																	

5.7. Develop a Single Object Maintenance User Interface

In the previous section, we created a search UI; now we'll proceed to create a single object maintenance UI for the project definition and WBS hierarchy.

<p>Return to the list of UI configurations and find the <i>USMD_OVP_GEN</i> Config.</p> <p>Choose <i>Copy</i>.</p>	<p>SAP Manage UI Configurations</p> <p>Copy</p> <p>View: [Standard View] Print Version Export</p> <table border="1"><thead><tr><th>Data Model</th><th>Application</th><th>Application Configuration</th></tr></thead><tbody><tr><td>BS_OVP_CC</td><td>BS_OVP_CC</td><td></td></tr><tr><td>USMD_ENTITY_VALUE2</td><td>USMD_ENTITY_VALUE2</td><td></td></tr><tr><td>USMD_OVP_GEN</td><td>USMD_OVP_GEN_TEMPLATE</td><td></td></tr><tr><td>USMD_SEARCH</td><td>USMD_SEARCH_TEMPLATE</td><td></td></tr><tr><td>USMD_SEARCH</td><td>USMD_SEARCH_TEMPLATE_02</td><td></td></tr><tr><td>OG</td><td>MDGF_OVP_GEN</td><td>MDGF_OG_OVP_BDC</td></tr><tr><td>OG</td><td>MDGF_OVP_GEN</td><td>MDGF_OG_OVP_BDCSET</td></tr></tbody></table>	Data Model	Application	Application Configuration	BS_OVP_CC	BS_OVP_CC		USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2		USMD_OVP_GEN	USMD_OVP_GEN_TEMPLATE		USMD_SEARCH	USMD_SEARCH_TEMPLATE		USMD_SEARCH	USMD_SEARCH_TEMPLATE_02		OG	MDGF_OVP_GEN	MDGF_OG_OVP_BDC	OG	MDGF_OVP_GEN	MDGF_OG_OVP_BDCSET
Data Model	Application	Application Configuration																							
BS_OVP_CC	BS_OVP_CC																								
USMD_ENTITY_VALUE2	USMD_ENTITY_VALUE2																								
USMD_OVP_GEN	USMD_OVP_GEN_TEMPLATE																								
USMD_SEARCH	USMD_SEARCH_TEMPLATE																								
USMD_SEARCH	USMD_SEARCH_TEMPLATE_02																								
OG	MDGF_OVP_GEN	MDGF_OG_OVP_BDC																							
OG	MDGF_OVP_GEN	MDGF_OG_OVP_BDCSET																							
<p>As before, the first thing we should do is change the affixes.</p> <p>As before, the name is as our data model.</p>	<p>Change Affixes of Target Configuration IDs</p> <p>Prefix: ZX</p> <p>Suffix: ZX X</p>																								
<p>Choose OK.</p> <p>As before, rename <i>Target Configuration ID</i> as shown.</p> <p>Deselect the <i>Copy</i> checkbox for <i>Control-Based Application</i> and start <i>Deep-Copy Mode</i>.</p>	<p>SAP Floorplan Manager: Application Hierarchy Browser - Application: USMD_OVP_GEN</p> <p>Browser Mode Deep-Copy Mode</p> <p>Application Hierarchy: Configuration Level Application Hierarchy: Context Based Adaptation Level</p> <p>Configurable Components All Change Affixes Start Deep-Copy Test</p> <table border="1"><thead><tr><th>Hierarchy</th><th>Copy</th><th>Component</th><th>Interface...</th><th>Configuration ID</th><th>Target Configuration ID</th></tr></thead><tbody><tr><td>Application Configuration</td><td><input checked="" type="checkbox"/></td><td></td><td></td><td>USMD_OVP_GEN_TEMPLATE</td><td>ZX_USMD_OVP_COMP_POSID</td></tr><tr><td>Context-Based Adaptation</td><td><input type="checkbox"/></td><td>FPM_ADAPTABLE_O...</td><td></td><td>USMD_OVP_CBA</td><td></td></tr><tr><td>Overview Page Floorplan</td><td><input type="checkbox"/></td><td>FPM_OVP_COMPO...</td><td></td><td>USMD_GEN_OVP_TEMPLATE</td><td></td></tr></tbody></table>	Hierarchy	Copy	Component	Interface...	Configuration ID	Target Configuration ID	Application Configuration	<input checked="" type="checkbox"/>			USMD_OVP_GEN_TEMPLATE	ZX_USMD_OVP_COMP_POSID	Context-Based Adaptation	<input type="checkbox"/>	FPM_ADAPTABLE_O...		USMD_OVP_CBA		Overview Page Floorplan	<input type="checkbox"/>	FPM_OVP_COMPO...		USMD_GEN_OVP_TEMPLATE	
Hierarchy	Copy	Component	Interface...	Configuration ID	Target Configuration ID																				
Application Configuration	<input checked="" type="checkbox"/>			USMD_OVP_GEN_TEMPLATE	ZX_USMD_OVP_COMP_POSID																				
Context-Based Adaptation	<input type="checkbox"/>	FPM_ADAPTABLE_O...		USMD_OVP_CBA																					
Overview Page Floorplan	<input type="checkbox"/>	FPM_OVP_COMPO...		USMD_GEN_OVP_TEMPLATE																					

<p>Return to the UI configurations list and choose the hyperlink for your UI <code>ZX_USMD_OVP_COMP_POSID</code>.</p> <p>Change the <code>USM_OTC</code> to suit your business object</p>	 <p>(BCV_CONTEXT_KEY): MDGAF_SOM</p> <p>(CRTYPE):</p> <p>(FPM_IGNORE_WIRE_SOURCE): X</p> <p>(FPM_PAGING_ACTIVE):</p> <p>(MDG_HC_COLOR_SAVED):</p> <p>(MDG_HC_COLOR_UNSAVED):</p> <p>(MDG_HC_DISABLE):</p> <p>(USMD_OTC): ZCKPS</p> <p>(USMD_SEARCH_EDITION_MODE):</p>																	
<p>On the same page at the top we will see <i>Component Usage</i>.</p> <p>Select the configuration name <code>USMD_GEN_OVP_TEMPLATE</code>.</p>	 <p>File ▾ Edit ▾ View ▾ Other Functions ▾</p> <p>Hi, CHRIS</p> <p>Application Configuration: ZX_USMD_OVP_COMP_POSID</p> <table border="1"> <thead> <tr> <th colspan="5">Assign Configuration Name</th> </tr> <tr> <th>Component Usage</th> <th>Component</th> <th>Implementation</th> <th>Configuration Name</th> </tr> </thead> <tbody> <tr> <td>USMD_OVP_GEN</td> <td>FPM_ADAPTABLE_OVP</td> <td>FPM_ADAPTABLE_OVP</td> <td>USMD_OVP_CBA</td> </tr> <tr> <td>OVP</td> <td>FPM_OVP_COMPONENT</td> <td>FPM_OVP_COMPONENT</td> <td>USMD_GEN_OVP_TEMPLATE</td> </tr> </tbody> </table>	Assign Configuration Name					Component Usage	Component	Implementation	Configuration Name	USMD_OVP_GEN	FPM_ADAPTABLE_OVP	FPM_ADAPTABLE_OVP	USMD_OVP_CBA	OVP	FPM_OVP_COMPONENT	FPM_OVP_COMPONENT	USMD_GEN_OVP_TEMPLATE
Assign Configuration Name																		
Component Usage	Component	Implementation	Configuration Name															
USMD_OVP_GEN	FPM_ADAPTABLE_OVP	FPM_ADAPTABLE_OVP	USMD_OVP_CBA															
OVP	FPM_OVP_COMPONENT	FPM_OVP_COMPONENT	USMD_GEN_OVP_TEMPLATE															
<p>On your new page, in the left column rename the <i>Title</i> and <i>Page Id</i> to a name of your choice.</p>	 <table border="1"> <thead> <tr> <th>Page Type</th> <th>Page Id</th> <th>Title</th> </tr> </thead> <tbody> <tr> <td>Main Page</td> <td>MAIN</td> <td>Process Projects</td> </tr> </tbody> </table>	Page Type	Page Id	Title	Main Page	MAIN	Process Projects											
Page Type	Page Id	Title																
Main Page	MAIN	Process Projects																

5.7.1. Configure the Technical UIBB

We start the UI configuration of the main page by creating a technical UIBB. The technical UIBB is never shown in the UI, but it participates in the Floorplan Manager event loop. This UIBB will be the root UIBB in the wire schema. The UIBB will just contain the key fields of entity UIBB.

In the center of your page, choose the *Section* button then choose *UIBB* and choose *Form Component*.

The screenshot shows the SAP Fiori Launchpad interface. In the top navigation bar, there are tabs for 'Overview Page Schema', 'Toolbar Schema', and 'Wire Schema'. Below the tabs, there are buttons for '+ Page Master Area', '+ Section', and '+ UIBB'. The '+ UIBB' button is highlighted with a blue background. To the right of these buttons is a search bar with a magnifying glass icon and a dropdown arrow. Below the buttons is a table with two columns: 'Element' and 'Config ID'. The 'Element' column contains several component names, and the 'Config ID' column is empty. A vertical scroll bar is visible on the right side of the table.

Element	Config ID
Analytics Chart Component	
Analytics List Component	
Attribute Filter Component	
BICS Component	
Carousel Component	
Chart Component	
Composite Component	
Form Component	
Form Repeater Component	

When opened fill in the details as shown

The screenshot shows the SAP Fiori Launchpad interface. In the top navigation bar, there are tabs for 'Overview Page Schema', 'Toolbar Schema', and 'Wire Schema'. Below the tabs, there are buttons for '+ Page Master Area', '+ Section', '+ UIBB', and a trash can icon. The '+ UIBB' button is highlighted with a blue background. To the right of these buttons is a search bar with a magnifying glass icon and a dropdown arrow. Below the buttons is a table with four columns: 'Element', 'Component', 'Config ID', and 'Window Name'. The 'Element' column shows 'Section: SECTION_1' expanded to show 'UIBB: Technical UIBB'. The 'Component' column shows 'FPM_FORM_UIBB_GL2'. The 'Config ID' column shows 'ZX_FORM_PSPID_TECHNICAL' with a red box around it. The 'Window Name' column shows 'FORM_WINDOW'. A vertical scroll bar is visible on the right side of the table.

Element	Component	Config ID	Window Name
Section: SECTION_1			
UIBB: Technical UIBB	FPM_FORM_UIBB_GL2	ZX_FORM_PSPID_TECHNICAL	FORM_WINDOW

Ensure you fill out the fields in the right column also.

The config ID must match the name of the config ID in the previous step.

We have enabled the *Hidden Element* attribute that makes this UIBB hidden but still participates in the event loop.

Standard Attributes of UIBB: Technical UIBB

* Component:	<input type="text" value="FPM_FORM_UIBB_GL2"/>
* Window Name:	<input type="text" value="FORM_WINDOW"/>
Config ID:	<input style="border: 2px solid red;" type="text" value="ZX_FORM_PSPID_TECHNICAL"/>
Instance ID:	<input type="text"/>
Column:	<input type="text" value="1"/>
Sequence Index:	<input type="text" value="1"/>
Hidden Element:	<input type="text" value="Hidden but Processed in Eve..."/>
Rendering Type:	<input type="text" value="With Panel"/>
Collapsed:	<input type="checkbox"/>
Title:	<input type="text" value="Technical UIBB"/>
Tooltip:	<input type="text"/>
Image:	<input type="text"/>
Padding:	<input type="text" value="Automatic (Default)"/>
Explanation Text:	<input type="text"/>

Return to the center column.

Change the view from *Overview Page Schema* to *Wire Schema*.

Here we will create a wire, which has the technical UIBB as the target UIBB but doesn't have a source UIBB. This makes it the root UIBB, which can be instantiated independently. Choose the *+ Wire* button.

The component is automatically filled. It is important you fill out the config ID with the same name in the previous steps.

Save your changes.

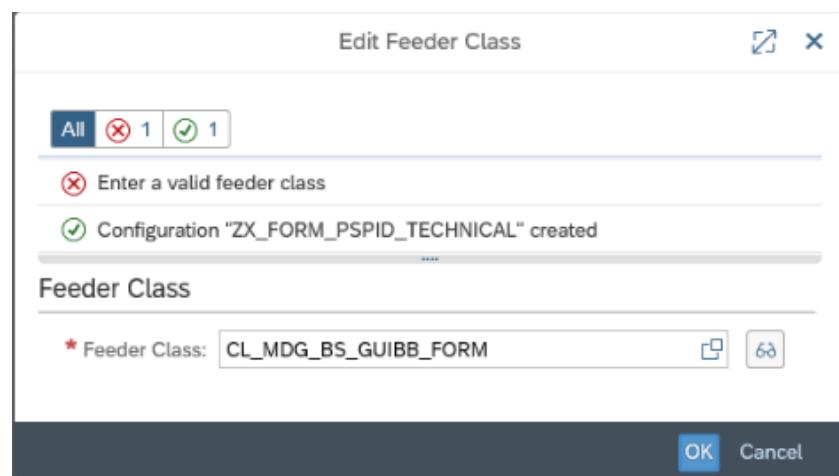
Return to the *Overview Page Schema* view and choose *Configure UIBB* in the top right corner.

The screenshot shows the 'Wire Schema' tab selected in a software interface. At the top, it displays the transaction 'MDG BOL Transaction Handler (CL_MDG_BS_BOL_TRANSACTION)'. Below this is a toolbar with buttons for '+ Wire' and 'Graphical Wire Editor'. A message in the center of the screen says 'The table does not contain any data'. There is a table with columns: Element, Component, Config ID, and Port Type. The first row is a header, and the second row contains a note: '(i) The table does not contain any data'.

The screenshot shows the 'Wire Schema' tab selected. The transaction 'MDG BOL Transaction Handler (CL_MDG_BS_BOL_TRANSACTION)' is still displayed. The table now contains one row: 'Element' (checkbox checked), 'Component' (checkbox checked, showing 'Wire: Form ZX_FORM_FPM_FORM_UIBB_GL2'), 'Config ID' (checkbox checked, showing 'ZX_FORM_PSPID_TECHNIC'), and 'Source Compo...' (checkbox checked). The 'Element' and 'Component' columns have icons indicating they are dropdowns.

The screenshot shows the 'Configure UIBB' dialog box. It has a table with two rows. The first row has columns 'Config ID' and 'Window Name'. The second row has columns 'ZX_FORM_PSPID_TECHNICAL' (highlighted with a red border) and 'FORM_WINDOW'. The 'Config ID' column is also highlighted with a red border.

When you come to the feeder screen, insert the *Feeder Class* as shown and choose *OK*.



Open the search in the component parameter and select *ZSP_ZX*.

Items (9)		
	Component	Name
<input type="radio"/>	ZHP_Y0	MDG Hierarchy Processing for Y0
<input type="radio"/>	ZHP_Z0	MDG Hierarchy Processing for Z0
<input type="radio"/>	ZHP_ZX	MDG Hierarchy Processing for ZX
<input type="radio"/>	ZMP_Y0	MDG Multi-Record Processing for Y0
<input type="radio"/>	ZMP_Z0	MDG Multi-Record Processing for Z0
<input type="radio"/>	ZMP_ZX	MDG Multi-Record Processing for ZX
<input type="radio"/>	ZSP_Y0	MDG Single Processing for Y0
<input type="radio"/>	ZSP_Z0	MDG Single Processing for Z0
<input checked="" type="radio"/>	ZSP_ZX	MDG Single Processing for ZX

Insert *PSPID* as the *Object Name* and it is important to always select the *Editable* checkbox.

Parameters																																																															
* Component:	<input type="text" value="ZSP_ZX"/>																																																														
* Object Name:	<input type="text" value="PSPID"/>																																																														
Editable:	<input checked="" type="checkbox"/>																																																														
Join Structure																																																															
<input type="button" value="Insert Child"/> <input type="button" value="Delete"/> <table border="1"> <tr> <td><input type="checkbox"/></td> <td>Join S...</td> <td>* Relat...</td> <td>* Com...</td> <td>* Assi...</td> <td>Suffix</td> <td>Dark</td> <td>Port Id...</td> </tr> <tr><td><input type="checkbox"/></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td><input type="checkbox"/></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td><input type="checkbox"/></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td><input type="checkbox"/></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td><input type="checkbox"/></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td><input type="checkbox"/></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>								<input type="checkbox"/>	Join S...	* Relat...	* Com...	* Assi...	Suffix	Dark	Port Id...	<input type="checkbox"/>								<input type="checkbox"/>								<input type="checkbox"/>								<input type="checkbox"/>								<input type="checkbox"/>								<input type="checkbox"/>							
<input type="checkbox"/>	Join S...	* Relat...	* Com...	* Assi...	Suffix	Dark	Port Id...																																																								
<input type="checkbox"/>																																																															
<input type="checkbox"/>																																																															
<input type="checkbox"/>																																																															
<input type="checkbox"/>																																																															
<input type="checkbox"/>																																																															
<input type="checkbox"/>																																																															

On the new screen ensure that *Project definition* is the only element. You can remove everything else.

Make sure to save. Return by choosing the hyperlink as before:

USMD_GEN_OVP_TEMPLATE

USMD_OVP_GEN: ZX_USMD_OVP_COMP_POSID > OVP: USMD_GEN_OVP_TEMPLATE > Form UIBB: ZX_FORM_PSPID_TECHNICAL													
<p>▼ Preview + Element ▾ + Group + Line </p> <table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td></tr></table>	A	B	C	D	E	F	G	H	I	J	K	L	M
A	B	C	D	E	F	G	H	I	J	K	L	M	

5.7.2. Configure Form UIBB for Project Definition.

Here we will add the first visible UIBB. This UIBB will be a form and will hold the project definition. We will also configure a wire with port type lead from the technical UIBB to the project definition UIBB.

Create a new UIBB in 5.7.1
Configure the Technical
UIBB. Again, create a form
component UIBB.

✓ Overview Page Schema Toolbar Schema Wire Schema

+ Page Master Area + Section + UIBB ▾

Element	Config ID	Window Name
Analytics Chart Component		
Analytics List Component		
Attribute Filter Component		
BICS Component		
Carousel Component		
Chart Component		
Composite Component		
Form Component	ZX_FORM_PSPID_TECHNICAL	FORM_WINDOW
Form Repeater Component		

As before, enter a configuration name.

Overview Page Schema		Toolbar Schema	Wire Schema
+ Page Master Area + Section + UIBB ▾ trash ^ ▼			
Element	Component	Config ID	
<input type="checkbox"/>	Section: SECTION_1		
<input type="checkbox"/>	UIBB: Technical UIBB	FPM_FORM_UIBB_GL2	ZX_FORM_PSPID_TECHNICAL
<input checked="" type="checkbox"/>	UIBB	FPM_FORM_UIBB_GL2	ZX_FORM_PSPID

Entering the details in the right column ensure we write the configuration name the same. Note this time we have chosen the *Hidden Element* as *Visible*.

Standard Attributes of UIBB

* Component:	FPM_FORM_UIBB_GL2
* Window Name:	FORM_WINDOW
Config ID:	ZX_FORM_PSPID
Instance ID:	
Column:	1
Sequence Index:	2
Hidden Element:	Visible
Rendering Type:	With Panel
Collapsed:	<input type="checkbox"/>
Title:	Project definition

As before, create a wire and enter the same *Config ID* as in the last step.

The source config name will be the name of the first wire we created

ZX_FORM_PSPID_TECHNICAL.

Change the *Port Type* to *Lead Selection*.

In the right column ensure the details are filled out as the same.

Input the *Port Identifier* and *Connector Class*:

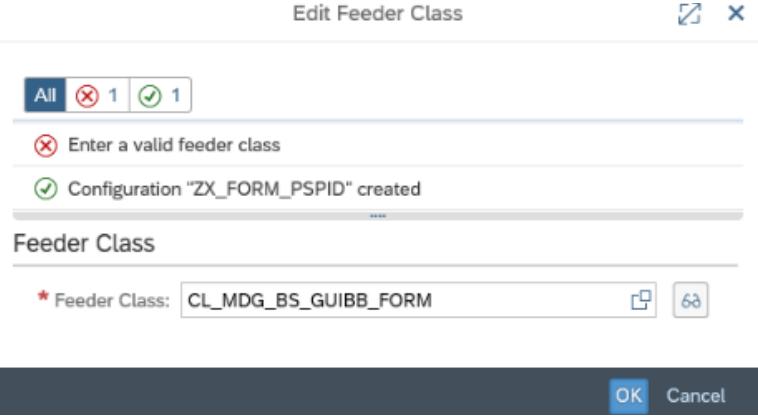
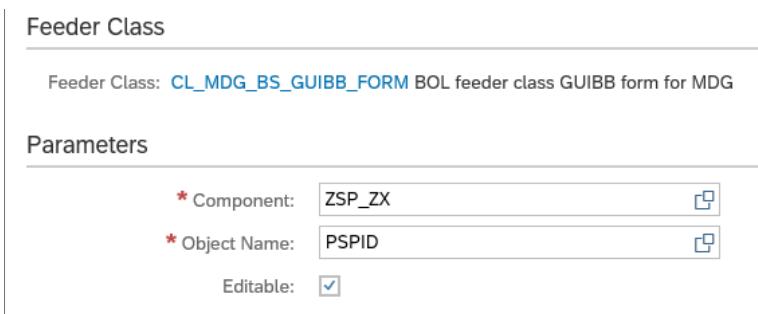
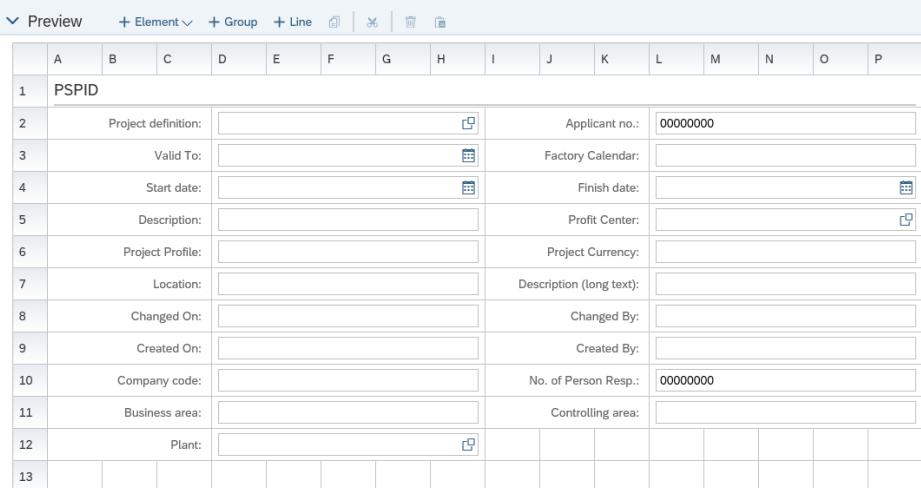
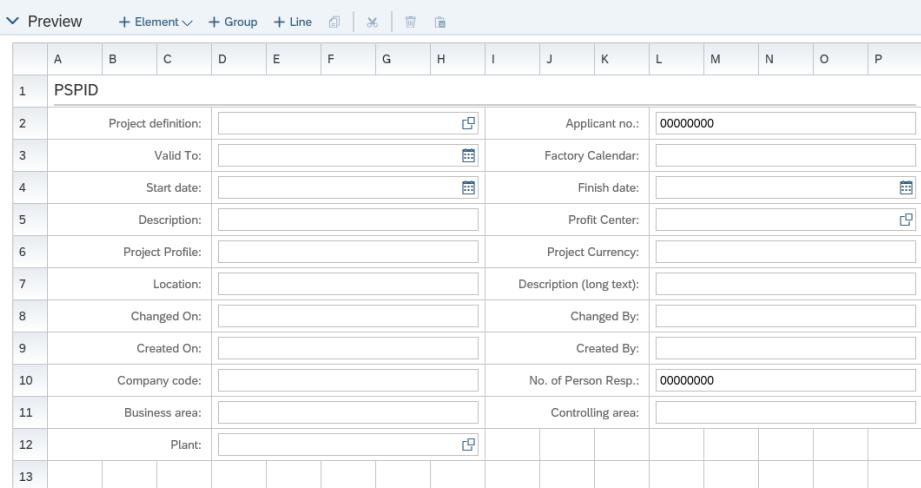
CL_FPM_CONNECTOR_BOL_IDENTITY.

Wire Schema						
Transaction: MDG BOL Transaction Handler (CL_MDG_BS_BOL_TRANSACTION)						
Graphical Wire Editor						
Element	Component	Config ID	Source Component	Source Config Name	Port Type	
Wire: Form ZX_FORM_FPM_FORM_UIBB_GL2	ZX_FORM_PSPID_TECHNICAL					
Wire: Form ZX_FORM_FPM_FORM_UIBB_GL2	ZX_FORM_PSPID	FPM_FORM_UIBB_GL2	ZX_FORM_PSPID_TECH	Lead Sele		

Attributes General Settings

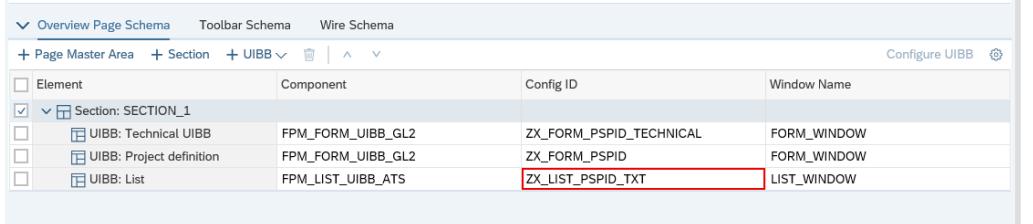
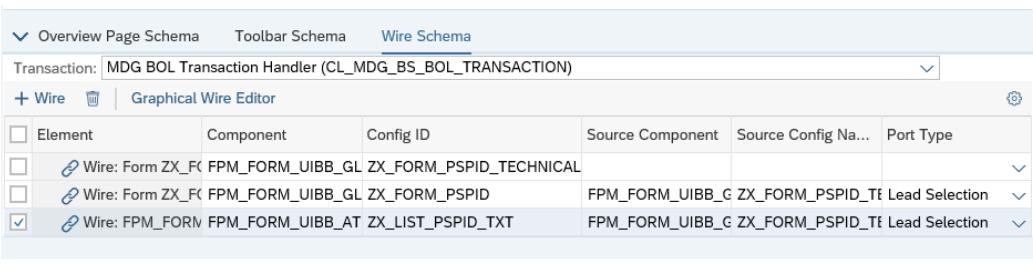
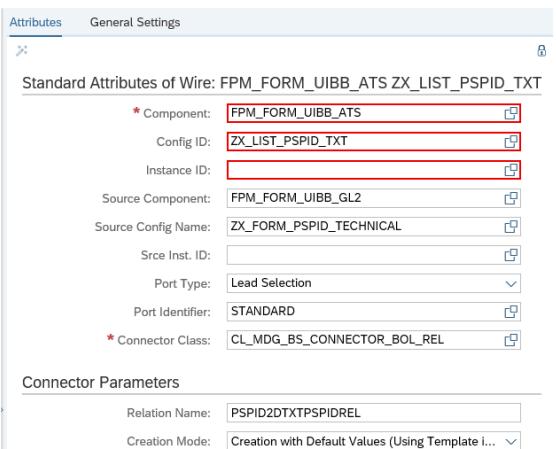
Standard Attributes of Wire: Form ZX_FORM_PSPID

* Component:	FPM_FORM_UIBB_GL2
Config ID:	ZX_FORM_PSPID
Instance ID:	
Source Component:	FPM_FORM_UIBB_GL2
Source Config Na...:	ZX_FORM_PSPID_TECHNICAL
Srce Inst. ID:	
Port Type:	Lead Selection
Port Identifier:	STANDARD
* Connector Class:	CL_FPM_CONNECTOR_BOL_I

Save your changes.	
As before, choose the <i>Configure UIBB</i> . We will be using the same feeder class in this case.	
Input the details the same as the previous UIBB.	
This time we can have more elements.	
Arrange your elements as shown in the picture.	
Save and return as previous.	

5.7.3. Configure the List UIBB for Language-Dependent Texts

The project definition (PSPID) has language-dependent texts activated, allowing the description for the project definition to be maintained in multiple languages. To enable this functionality in the UI, you need to add a list UIBB.

<p>Create a new UIBB as in section 5.7.1 Configure the Technical UIBB except this time select <i>List Component</i> instead of <i>Form Component</i>. Enter the config ID and ensure the details in the right column are entered correctly.</p>	
<p>Create a new wire. Insert a <i>Config ID</i> and name the other columns as before.</p>	
<p>In the right column enter the details as shown. Note we are using a different <i>Connector Class</i> this time.</p> <p>Enter the connector parameters as shown.</p>	

As before, choose *Configure UIBB*. This time we are using a different feeder class.

Insert the feeder class shown.

Edit Feeder Class

All 2

Feeder Class

* Feeder Class: `CL_MDG_BS_GUIBB_LIST`

OK Cancel

Again, choose the component as `ZSP_ZX`.

Insert the object name as `DTxPSPID`.

Edit Parameters

Feeder Class

Feeder Class: `CL_MDG_BS_GUIBB_LIST` BOL feeder class GUIBB list for MDG

Parameters

* Component: `ZSP_ZX`
* Object Name: `DTxtPSPID`
Editable:

Join Structure

Insert Child

	Join...	* Rela...	* Co...	* Assi...	Suffix	Dark	Port I...
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							

Arrange the elements as shown in the picture.

Any additional elements can be removed.

Preview

New IS_ENTITY_DELETED DISCARD_DELETION IS_DISCARD_DEL_ALLOWED Copy New

Actions	Language Key	Description (long text)	Language Key Description

List UIBB Schema Toolbar Schema Menu Schema Quickview Schema

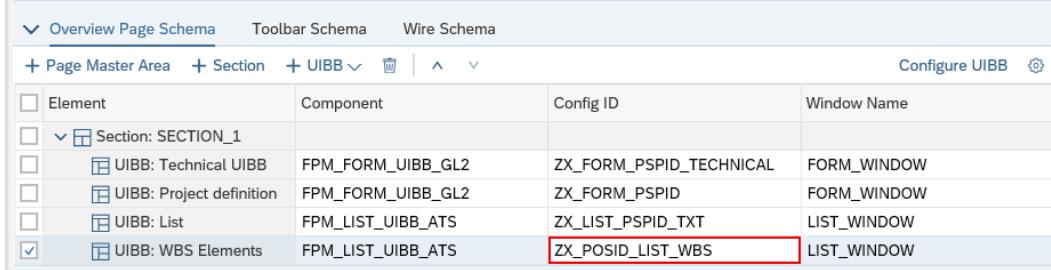
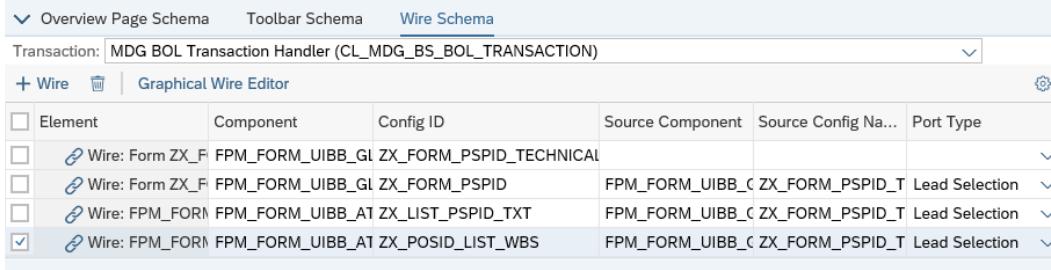
+ Column + Multi Editor Column + Header Group

Element	Display Type	Header	Tooltip	Header Tooltip
Column: FPM_ROW_ACTIC	Link To Action			
Column: LANGU	Input Field	Language Key		
Column: TXTLG	Input Field	Description (long text)		
Column: LANGU_TXT	Input Field	Language Key Description		

Save your changes.

5.7.4. Configure the List UIBB for Creating and Changing WBS elements

A project definition can have multiple WBS elements assigned to it. Each WBS element has its own set of attributes and is assigned to the WBS hierarchy. To meet these requirements, we first create a list UIBB in the main page and then create an edit page; this edit page is tagged as the default edit page for the list UIBB.

<p>As before, create a list component UIBB.</p> <p>Name as shown and ensure the right column is entered correctly.</p>	 <table border="1"> <thead> <tr> <th>Element</th><th>Component</th><th>Config ID</th><th>Window Name</th></tr> </thead> <tbody> <tr> <td>Section: SECTION_1</td><td></td><td></td><td></td></tr> <tr> <td>UIBB: Technical UIBB</td><td>FPM_FORM_UIBB_GL2</td><td>ZX_FORM_PSPID_TECHNICAL</td><td>FORM_WINDOW</td></tr> <tr> <td>UIBB: Project definition</td><td>FPM_FORM_UIBB_GL2</td><td>ZX_FORM_PSPID</td><td>FORM_WINDOW</td></tr> <tr> <td>UIBB: List</td><td>FPM_LIST_UIBB_ATS</td><td>ZX_LIST_PSPID_TXT</td><td>LIST_WINDOW</td></tr> <tr> <td>UIBB: WBS Elements</td><td>FPM_LIST_UIBB_ATS</td><td>ZX_POSID_LIST_WBS</td><td>LIST_WINDOW</td></tr> </tbody> </table>	Element	Component	Config ID	Window Name	Section: SECTION_1				UIBB: Technical UIBB	FPM_FORM_UIBB_GL2	ZX_FORM_PSPID_TECHNICAL	FORM_WINDOW	UIBB: Project definition	FPM_FORM_UIBB_GL2	ZX_FORM_PSPID	FORM_WINDOW	UIBB: List	FPM_LIST_UIBB_ATS	ZX_LIST_PSPID_TXT	LIST_WINDOW	UIBB: WBS Elements	FPM_LIST_UIBB_ATS	ZX_POSID_LIST_WBS	LIST_WINDOW						
Element	Component	Config ID	Window Name																												
Section: SECTION_1																															
UIBB: Technical UIBB	FPM_FORM_UIBB_GL2	ZX_FORM_PSPID_TECHNICAL	FORM_WINDOW																												
UIBB: Project definition	FPM_FORM_UIBB_GL2	ZX_FORM_PSPID	FORM_WINDOW																												
UIBB: List	FPM_LIST_UIBB_ATS	ZX_LIST_PSPID_TXT	LIST_WINDOW																												
UIBB: WBS Elements	FPM_LIST_UIBB_ATS	ZX_POSID_LIST_WBS	LIST_WINDOW																												
<p>Create a wire as before, with the same <i>Source Component</i> and <i>Source Config Name</i>.</p>	 <table border="1"> <thead> <tr> <th>Element</th><th>Component</th><th>Config ID</th><th>Source Component</th><th>Source Config Na...</th><th>Port Type</th></tr> </thead> <tbody> <tr> <td>Wire: Form ZX_F</td><td>FPM_FORM_UIBB_GL</td><td>ZX_FORM_PSPID_TECHNICAL</td><td></td><td></td><td></td></tr> <tr> <td>Wire: Form ZX_F</td><td>FPM_FORM_UIBB_GL</td><td>ZX_FORM_PSPID</td><td>FPM_FORM_UIBB_C</td><td>ZX_FORM_PSPID_T</td><td>Lead Selection</td></tr> <tr> <td>Wire: FPM_FORM</td><td>FPM_FORM_UIBB_AT</td><td>ZX_LIST_PSPID_TXT</td><td>FPM_FORM_UIBB_C</td><td>ZX_FORM_PSPID_T</td><td>Lead Selection</td></tr> <tr> <td>Wire: FPM_FORM</td><td>FPM_FORM_UIBB_AT</td><td>ZX_POSID_LIST_WBS</td><td>FPM_FORM_UIBB_C</td><td>ZX_FORM_PSPID_T</td><td>Lead Selection</td></tr> </tbody> </table>	Element	Component	Config ID	Source Component	Source Config Na...	Port Type	Wire: Form ZX_F	FPM_FORM_UIBB_GL	ZX_FORM_PSPID_TECHNICAL				Wire: Form ZX_F	FPM_FORM_UIBB_GL	ZX_FORM_PSPID	FPM_FORM_UIBB_C	ZX_FORM_PSPID_T	Lead Selection	Wire: FPM_FORM	FPM_FORM_UIBB_AT	ZX_LIST_PSPID_TXT	FPM_FORM_UIBB_C	ZX_FORM_PSPID_T	Lead Selection	Wire: FPM_FORM	FPM_FORM_UIBB_AT	ZX_POSID_LIST_WBS	FPM_FORM_UIBB_C	ZX_FORM_PSPID_T	Lead Selection
Element	Component	Config ID	Source Component	Source Config Na...	Port Type																										
Wire: Form ZX_F	FPM_FORM_UIBB_GL	ZX_FORM_PSPID_TECHNICAL																													
Wire: Form ZX_F	FPM_FORM_UIBB_GL	ZX_FORM_PSPID	FPM_FORM_UIBB_C	ZX_FORM_PSPID_T	Lead Selection																										
Wire: FPM_FORM	FPM_FORM_UIBB_AT	ZX_LIST_PSPID_TXT	FPM_FORM_UIBB_C	ZX_FORM_PSPID_T	Lead Selection																										
Wire: FPM_FORM	FPM_FORM_UIBB_AT	ZX_POSID_LIST_WBS	FPM_FORM_UIBB_C	ZX_FORM_PSPID_T	Lead Selection																										
<p>In the right column enter the details as shown. Note we have changed the <i>Connector Class</i> again.</p> <p>The port identifier is the same.</p>																															

Save and choose <i>Configure UIBB</i> .	
Insert the feeder class. We are using the same <i>Feeder Class</i> as before.	
Enter the details as shown. Choose OK.	

In the right column you will see an action assignment.

Insert Events:

Show
FPM_CALL_DEFAULT_EDIT_PAGE.

Action Assignment

FPM Event ID	Image	Label	Hide Label	Tooltip
SHOW		Details	<input type="checkbox"/>	
FPM_CALL_DEFAULT_EDIT_PAGE	~Icon/Edit		<input type="checkbox"/>	
DISCARD_DELETE	~Icon/Undo		<input type="checkbox"/>	Discard Deletion

Arrange your UIBB layout so that it appears as shown in the picture.

The screenshot shows the SAP Fiori Launchpad configuration interface. At the top, there's a toolbar with buttons for 'New', 'IS_ENTITY_DELETED', 'DISCARD_DELETION', 'IS_DISCARD_DEL_ALLOWED', 'DISABLE_HIGHLIGHT_DEL', 'Copy', and download. Below the toolbar, the 'Preview' tab is selected, showing a table with columns: Actions, WBS element, Applicant no., Description, Project definition, and WBS element (Internal id). There are four rows in the table. Below the preview, the 'List UIBB Schema' tab is selected, showing a table of columns with their properties. One column, 'Column: POST1', is currently selected and highlighted in blue. The table includes columns for Element, Display Type, Header, Tooltip, and Header Tooltip. Other columns listed include FPM_ROW_A Link To Action, POSID, ASTNR, PROJ_DEF, and PS_POSID.

Save your changes

We have now fully implemented the UI for our custom data model.

5.8. Process Modelling

In MDG every change including the creation of master data is done through a change request, which is roughly a carrier of change to the master data. After change requests are initiated, they need to be processed by applying governance rules and collaboration.

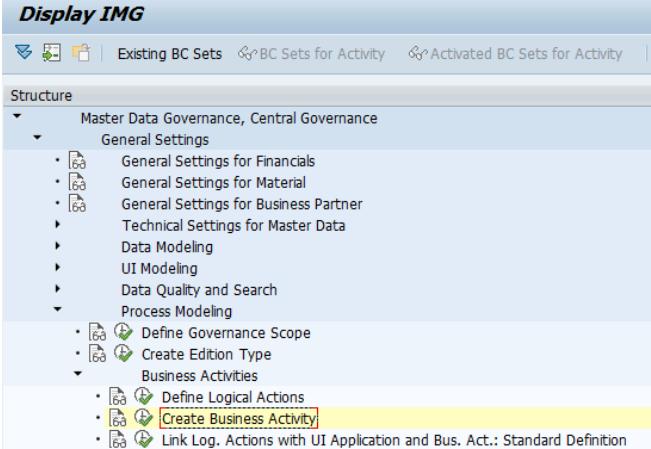
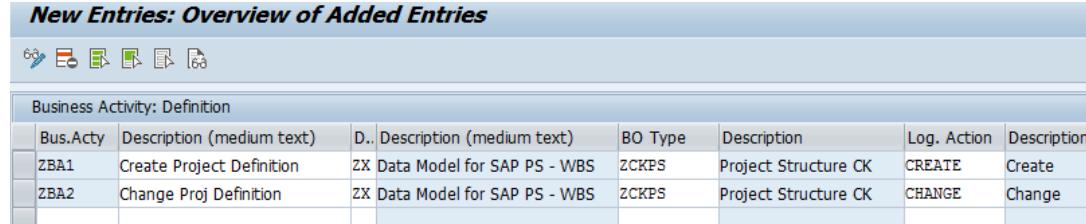
The process model provides the required input for change request creation and process of change requests. The process modelling configuration node in SAP MDG is a group of similar configuration activities required to execute the change request process.

For process modelling, the first step is to create a new business activity to tie an action with the business object and then maintain navigation settings of the custom UI application using business activities and actions.

5.8.1. Create a New Business Activity

To design a business activity, you first must understand what kind of actions you intend to perform and on which business object. Each combination of logical action and business object becomes one business activity. In our case, we will create two business activities.

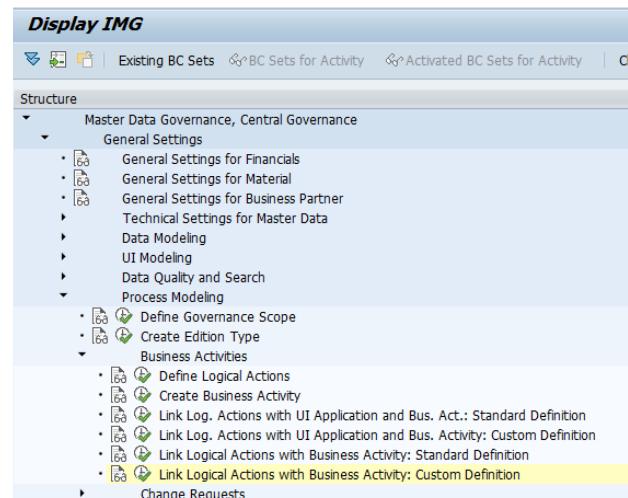
- ZBA1 Create project definition and WBS elements.
- ZBA2 Change project definition and assign WBS elements to the project hierarchy.

<p>In transaction MDGIMG, navigate to General Settings → Process Modeling → Business Activities → Create Business Activity.</p>																									
<p>Create two business activities.</p>	 <table border="1" data-bbox="425 1365 1511 1486"> <thead> <tr> <th>Bus. Acty</th> <th>Description (medium text)</th> <th>D.</th> <th>Description (medium text)</th> <th>BO Type</th> <th>Description</th> <th>Log. Action</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ZBA1</td> <td>Create Project Definition</td> <td>ZX</td> <td>Data Model for SAP PS - WBS</td> <td>ZCKPS</td> <td>Project Structure CK</td> <td>CREATE</td> <td>Create</td> </tr> <tr> <td>ZBA2</td> <td>Change Proj Definition</td> <td>ZX</td> <td>Data Model for SAP PS - WBS</td> <td>ZCKPS</td> <td>Project Structure CK</td> <td>CHANGE</td> <td>Change</td> </tr> </tbody> </table>	Bus. Acty	Description (medium text)	D.	Description (medium text)	BO Type	Description	Log. Action	Description	ZBA1	Create Project Definition	ZX	Data Model for SAP PS - WBS	ZCKPS	Project Structure CK	CREATE	Create	ZBA2	Change Proj Definition	ZX	Data Model for SAP PS - WBS	ZCKPS	Project Structure CK	CHANGE	Change
Bus. Acty	Description (medium text)	D.	Description (medium text)	BO Type	Description	Log. Action	Description																		
ZBA1	Create Project Definition	ZX	Data Model for SAP PS - WBS	ZCKPS	Project Structure CK	CREATE	Create																		
ZBA2	Change Proj Definition	ZX	Data Model for SAP PS - WBS	ZCKPS	Project Structure CK	CHANGE	Change																		

5.8.2. Assign Business Activities and Logical Actions to Business Objects

A logical action represents the operation to be performed on the master data by an actor in the process (for example, create, change, or delete). Business activities add business context to logical actions by linking them with business objects such as create supplier, change material, and delete account. Business activities are defined by assigning an action, data model, and business object.

In transaction MDGIMG, navigate to **General settings** → **Process Modelling** → **Business Activities** → **Link Logical Actions with Business Activity Custom Definition.**



In our case we don't need to create a new action but reuse an SAP delivered action such as change and create. To do this select the two actions with change and create and choose **Copy As**.

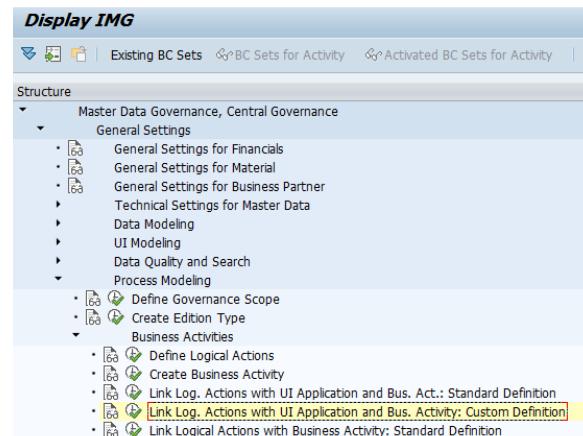
Change View "Business Activity: Determination": Overview						
New Entries New Edit Delete Copy Print Help						
Business Activity: Determination						
UI Application Name	UI Configuration	Log. Action	Description	Bu...	Description (medium text)	
MDG_BS_GEN_MC_O...	MDG_BS_GEN_MC_OVP_SF	MULTI	Multiple-Reco...	SFCM	Processing of Multiple Airlines	
MDG_BS_MAT_OVP	ZBS_MAT_OVP00	CREATE	Create	MAT1	Create Material	
USMD_OVP_GEN	USMD_SF_OVP_CARR_02	CHANGE	Change	SFC2	Change Airline	
USMD_OVP_GEN	USMD_SF_OVP_CARR_02	CREATE	Create	SFC1	Create Airline	
USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	DISPLAY	Display	ZBA2	Display Project Definition	

We can now edit those actions to suit our **UI Configuration** and **Business Activity**.

Change View "Business Activity: Determination": Overview of Selected S						
Business Activity: Determination						
UI Application Name	UI Configuration	Log. Action	Description	Bu...	Description (medium text)	
USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	CHANGE	Change	ZBA2	Change Proj Definition	
USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	CREATE	Create	ZBA1	Create Project Definition	

Creating cross-application navigation. First, like before access this through MDGIMG .

Navigate to **General Settings** → **Process Modelling** → **Business Activities** → **Link Log Actions with UI Application and Bus. Activity Custom Definition**.



Choose New Entries and enter the values shown.

This is the list of configurations created for cross-application navigation between the generic search UI and the custom object UI for actions create, change, and display.

New Entries: Overview of Added Entries



Maintenance Navigation

BO Type	Log. Action	Current UI Application Name	Current UI Configuration	Target UI Application Name	Target UI Configuration	Bus.Acty
ZCKPS	CHANGE	*	*	USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	ZBA2
ZCKPS	CHANGE	USMD_SEARCH	ZX_USMD_SEARCH	USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	ZBA2
ZCKPS	CREATE	*	*	USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	ZBA1
ZCKPS	CREATE	USMD_SEARCH	ZX_USMD_SEARCH	USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	ZBA1
ZCKPS	DISPLAY	*	*	USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	
ZCKPS	DISPLAY	USMD_SEARCH	ZX_USMD_SEARCH	USMD_OVP_GEN	ZX_USMD_OVP_COMP_POSID	

5.8.3. Create Change Request Type

The change request type is a key characteristic of the change request that determines how a change request is processed. The change request type links a change request to the workflow, data model and business activity.

Open Create Change Request Type through the transaction MDGIMG as shown.

Display IMG

Existing BC Sets BC Sets for Activity Activated BC Set

Structure

- Master Data Governance, Central Governance
 - General Settings
 - General Settings for Financials
 - General Settings for Material
 - General Settings for Business Partner
 - Technical Settings for Master Data
 - Data Modeling
 - UI Modeling
 - Data Quality and Search
 - Process Modeling
 - Define Governance Scope
 - Create Edition Type
 - Business Activities
 - Change Requests
 - Edit Statuses of Change Requests
 - Create Change Request Type**

Find the MAT01 change request and choose Copy.

Change View "Type of Change Request": Overview of Selected Set

Dialog Structure		Type of Change Request						
Type of Change Request		Type of Chg. Request	E...	Data Model	Description (medium text)	Main Entity Type	Workflow	Single Obj...
Entity Types	ZXWBSCRE	ZX	Create Projects	PSPID	WS60800086	<input checked="" type="checkbox"/>		

Then change the details as shown.

We will need to select all of the *Entity Types* and then delete them.

Change View "Entity Types": Overview					
Dialog Structure		Type of Chg. Request	Entity Types		
Type of Change Requests		ZXWBSCRE	Entity Types		
Entity Types			Entity Type	Scenario	Configuration ID
DRADBASIC			MATCHGMNG		
MATERIAL			MKALBASIC		

Choose New Entries and add the two *Entity Types* shown.

New Entries: Overview of Added Entries					
Dialog Structure		Type of Chg. Request	Entity Types		
Type of Change Requests		ZXWBSCRE	Entity Types		
Entity Types			Entity Type	Scenario	Configuration ID
PSPID					Optional
POSID					Message Output
<input checked="" type="checkbox"/>					Standard
<input checked="" type="checkbox"/>					Standard
<input checked="" type="checkbox"/>					Standard
<input checked="" type="checkbox"/>					Standard

Open business activities and as before, remove the activity associated with MAT01 and then add the business activity we created.

Change View "Business Activities": Overview					
Dialog Structure		Type of Chg. Request	Business Activities		
Type of Change Requests		ZXWBSCRE	Business Activities		
Entity Types			Bus. Activity	Description (medium text)	
Scope on Entity T			ZBA1	Create Project Definition	

We need to create a 2nd change request for change. We can simply copy the change request type we just created.

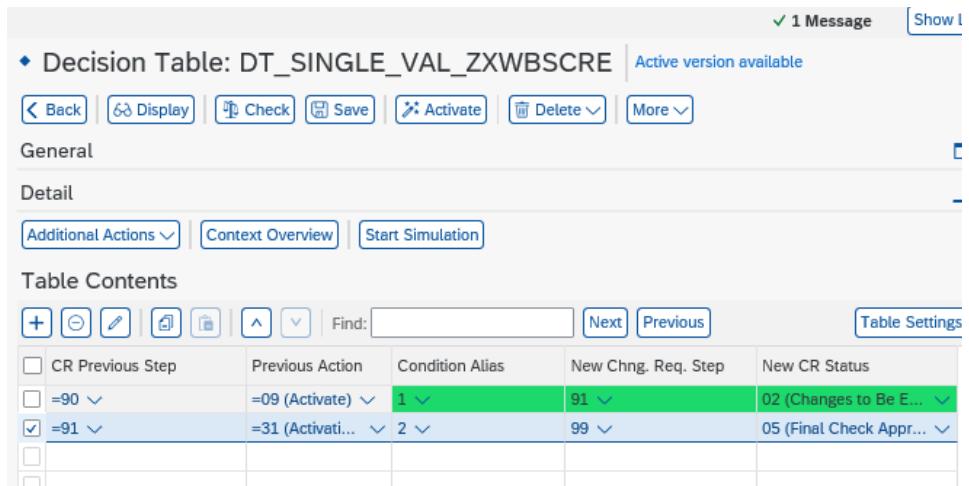
Change View "Type of Change Request": Overview				
Dialog Structure		Type of Change Request		
Type of Change Requests		Type of Chg. Request	Edition Type	Data Model
Entity Types		ZXWBSCRE	ZX	Create Projects

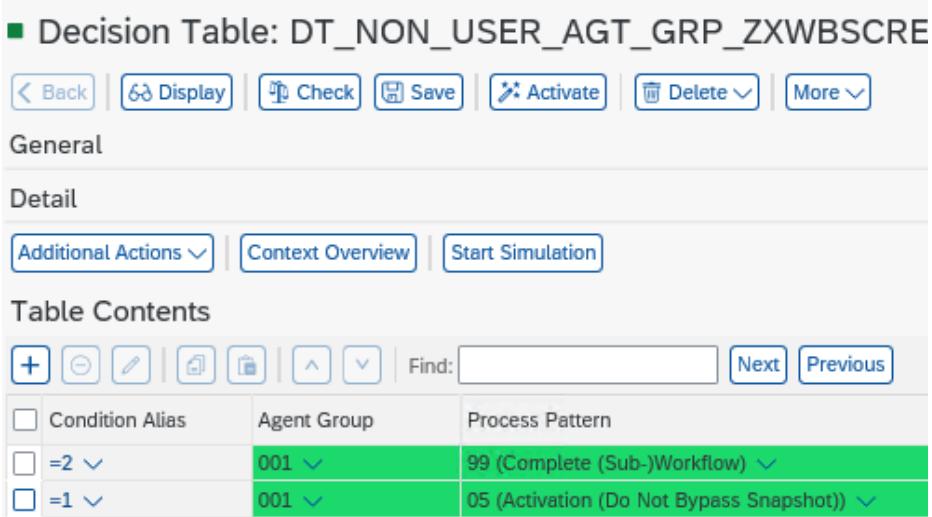
Change the details to as shown to associate it with changing projects.	
Change the business activity to the one we created for change projects.	
Save your changes.	

5.8.4. Create a Workflow

We need to assign a workflow to the change request type. In our case we will assign a rule-based workflow as it works with easily maintainable decision tables and is highly adaptable to customer needs.

As previously we will be opening this through the MDGIMG transaction. Open Configure Rule-Based Workflow as shown.	
---	--

Insert the create change request type we made as shown.	
Locate the decision tables and open the single value decision table.	
Include at least these two lines as shown. Extend it with your workflow steps.	
Save and activate.	

<p>Next open the non-user decision table.</p> <p>Include at least these two lines as shown.</p> <p>Extend it with your workflow steps.</p>	 <p>Decision Table: DT_NON_USER_AGT_GRP_ZXWBSCRE</p> <p>General</p> <p>Detail</p> <p>Table Contents</p> <table border="1"> <thead> <tr> <th>Condition Alias</th> <th>Agent Group</th> <th>Process Pattern</th> </tr> </thead> <tbody> <tr> <td>=2</td> <td>001</td> <td>99 (Complete (Sub-)Workflow)</td> </tr> <tr style="background-color: #00ffcc;"> <td>=1</td> <td>001</td> <td>05 (Activation (Do Not Bypass Snapshot))</td> </tr> </tbody> </table>	Condition Alias	Agent Group	Process Pattern	=2	001	99 (Complete (Sub-)Workflow)	=1	001	05 (Activation (Do Not Bypass Snapshot))
Condition Alias	Agent Group	Process Pattern								
=2	001	99 (Complete (Sub-)Workflow)								
=1	001	05 (Activation (Do Not Bypass Snapshot))								
Save and activate.										

5.8.5. Creating Our CR Wire

Now that we have created our change request and assigned a workflow, we must create a link between that and our UI. We have created our UI and our CR although as of now they are not linked, and our model won't work. We need to create a CR wire to do this in our UI configurations.

<p>Locate the UI configurations page as we did earlier and find our UI configuration.</p> <p>Choose <i>Details</i>.</p> <p>Then choose <i>Change</i>.</p>	 <p>ZX_USMD_OVP_COMP_POSID</p> <p>ZX_USMD_SEARCH</p> <p>USMD_GEN_OVP_TEMPLATE</p> <p>USMD_SEARCH_OVP_TEMPLATE</p> <p>Details</p> <p>Details</p>
<p>We need to create a wire to connect our change request with our UI.</p> <p>Choose <i>New</i> and then <i>crWires</i>.</p>	 <p>Component-Defined</p> <p>You can insert elements using the context menu for table rows</p> <p>Configuration Context</p> <p>New</p> <p>searchUiBibs</p> <p>crWires</p> <p>settings</p>

Fill in the details as shown.

As you can see the *Page Id* represents the page we created earlier.

The source config name is our technical UIBB.

* Page Id:	MAIN	<input checked="" type="checkbox"/> Final
connector:	CL_MDG_BS_CONNECTOR_BOL_CR_REL	<input type="checkbox"/> Final
Port Type:	Lead Selection	<input type="checkbox"/> Final
Port Identifier:	STANDARD	<input type="checkbox"/> Final
Source Component:	FPM_FORM_UIBB_GL2	<input type="checkbox"/> Final
Source Config Name:	ZX_FORM_PSPID_TECHNICAL	<input type="checkbox"/> Final
Src Config Type:	General	<input type="checkbox"/> Final
Src Config Ver:		<input type="checkbox"/> Final
Srce Inst. ID:		<input type="checkbox"/> Final

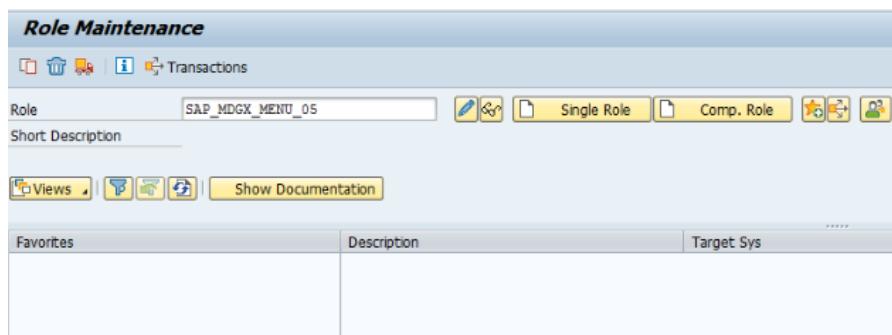
5.8.6. Creating Our Role

Roles are used for configuring authorization profiles and menus for users. Roles are directly assigned to the user master and can also be used to configure personalization values for users. The role for menu is used exclusively to configure the menu entries in SAP Business Client.

Search transaction PFCC.

Enter the role
SAP_MDGX_MENU_05

Then choose *Copy* in the top left corner.



Then enter the name of your role and choose OK.



Enter a *Description* for your role.

Change Roles

Role: ZSAP_MGGZX_MENU_04 Obsolete: Role Documentation:

Description: Master Data Governance for Custom Objects: PS

Target System: No destination:

Personalization:

Administration Information:

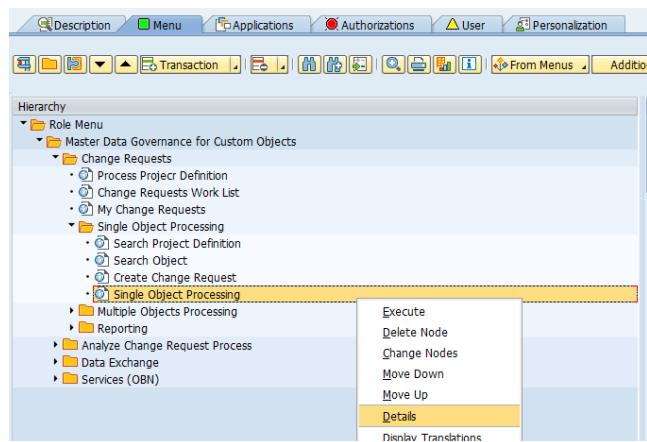
User: I352778	Created: 04.03.2021 16:42:07	Changed: 04.03.2021 16:43:22
Date: 04.03.2021		
Time: 16:42:07		

Transaction Inheritance: Derive from Role:

Save your changes.

Choose the *Menu* tab and then open *Single Object Processing*.

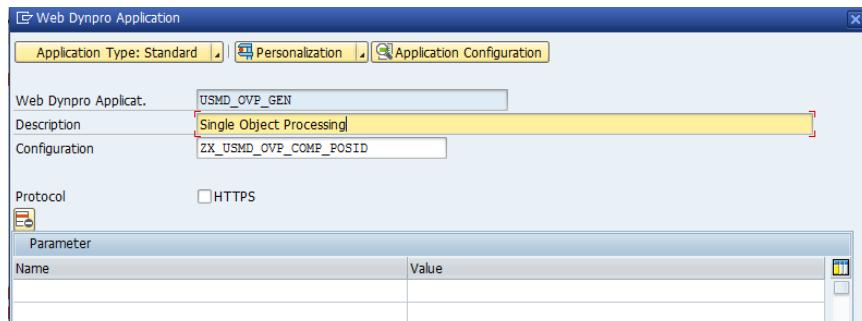
Open the context menu of *Single Object Process* and choose *Details*.



Enter the *Configuration ID* as the ID we created. You can search this to ensure you select the correct one.

Configuration ID (2) 7 Entries found	
Restrictions	
Configuration ID	Description
USMD_OVP_GEN_TEMPLATE	Template Configuration for Generic MDG Single-Object UI
USMD_SF_OVP_CARR	OVP for SF-CARR Airlines
USMD_SF_OVP_CARR_02	OVP for SF-CARR Airlines with Hry. Assignment
USMD_SF_OVP_PFLI	OVP for SF-PFLI Flight Connections
Y0_OVP_COUNTRY	Overviewpage for Y0 Country
Y0_OVP_COUNTRY_RS	Overviewpage for Y0 Country
ZX_USMD_OVP_COMP_POSID	Template Configuration for Generic MDG Single-Object UI

You can change the *Description* or leave as is then choose *OK*.



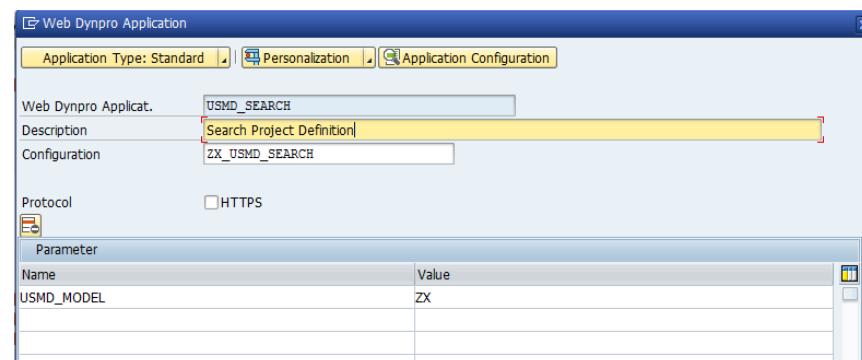
Like before open the *Single Object Processing* folder, then open the context menu on *Search Object* and choose *Details*.

Change the configuration to the search config you created.

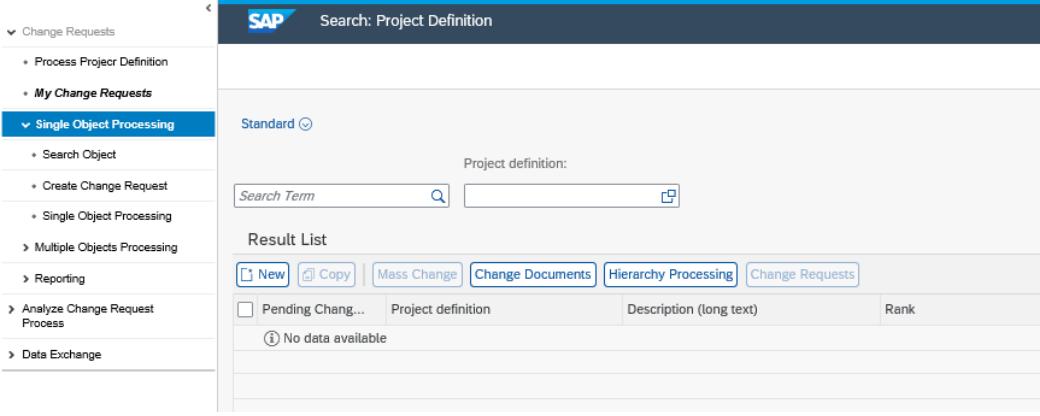
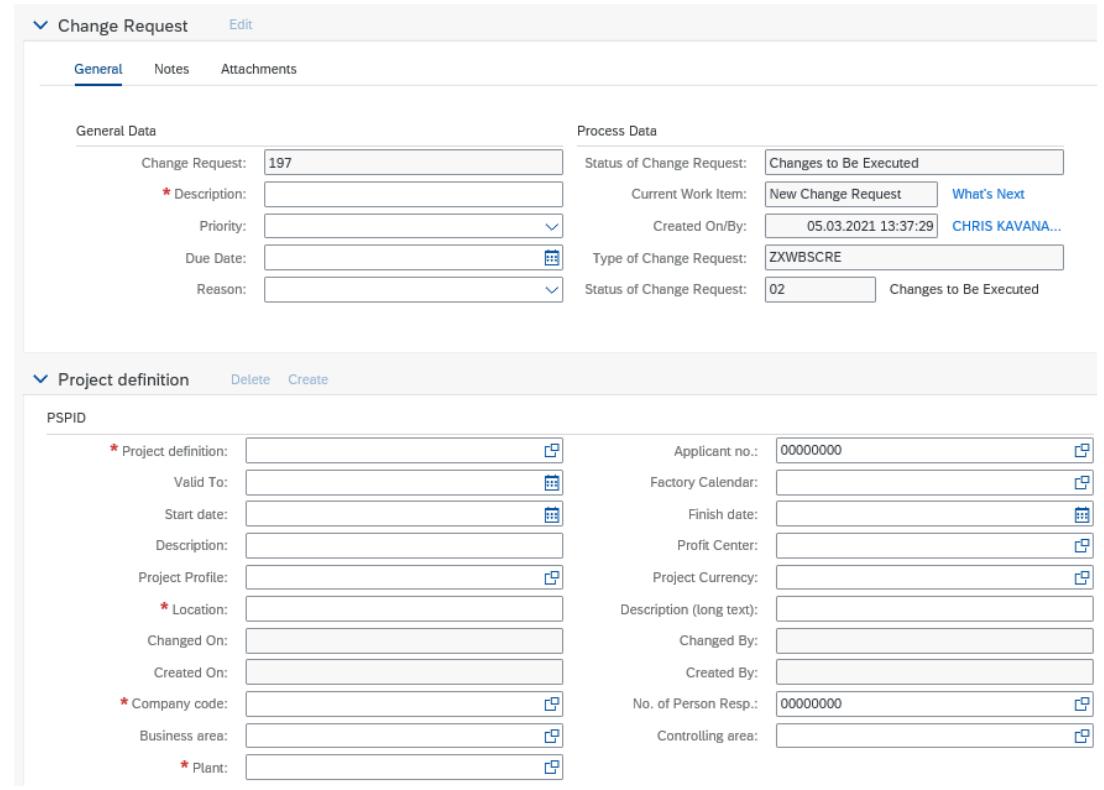
You can change your *Description* or leave as is.

Input your model as shown.

Save your role.



6. Testing our Data Model

<p>Open Transaction NWBC.</p> <p>Scroll down and choose the role we created.</p>	<ul style="list-style-type: none"> SAP MDG SUPPLIER APP SAP MDG WF ADM ZSAP_MGGZX_MENU_04 	<p>SAP Fiori : Supplier approval Master Data Governance: Authorizations for Workflow Batch User Master Data Governance for Custom Objects: PS</p>
<p>Choose Single Object Processing as you can see, we will be brought to our search screen.</p> <p>Choose New to create a new PS.</p>		
<p>Next, we will see the change request header and the UIBB for Project definition, which we created.</p>		

We can additionally see WBS elements option.

WBS Elements		Delete	New	Copy	New
POSID					
Actions	WBS element	Applicant no.	Description		
No data available					

7. Additional Information

7.1. Further Reading

7.1.1. Information on SAP MDG on SAP S/4HANA

- Exchange knowledge: [SAP Community](#) | [Q&A](#) | [Blog](#)
- Try SAP Master Data Governance on S/4HANA for free: [Trial Version](#)
- Try SAP Master Data Governance on S/4HANA on the SAP Cloud Appliance Library: [S/4HANA 2022 FPS1](#)
- Learn more: [Latest Release](#) | [Help Portal](#) | [How-to Information](#) | [Key Presentations](#)

7.1.2. SAP Roadmap Explorer

- Please see the [roadmap for SAP Master Data Governance](#)

7.1.3. Related Information

- Learn more: [Floorplan Manager for Web Dynpro ABAP](#) | [How to Adapt FPM](#) | [FPM Blog](#) | [How-to Information](#) | [Service Mapping Tool](#) | [SAP S/4HANA Cookbook CVI](#)

7.2. SAP Notes

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

Note	Description
1619534	How to Create, Enhance and Adapt FPM Applications
1637249	MDG: Information for efficient message processing
2105467	MDG Performance
2561461	Scope of support for SAP Master Data Governance (MDG)
1637249	MDG: Information for efficient message processing