

**SAP S/4 HANA MASTER DATA GOVERNANCE**

*Customer*

**2021-02**

MASTER DATA GOVERNANCE, MATERIAL

*DEMO EXPERIENCE*

**Creation of a Custom Data Model**



[Creating a Custom Data Model 3](#_Toc64972420)

[Introduction 3](#_Toc64972421)

[**Basic Data Model for Material** 3](#_Toc64972422)

[**Storage Types** 3](#_Toc64972423)

[**Relationship type** 3](#_Toc64972424)

[**Business Scenario** 4](#_Toc64972425)

[Step by Step Procedure 5](#_Toc64972426)

[**Creating Custom Elements** 5](#_Toc64972427)

[**Creating our Custom Data Model** 10](#_Toc64972428)

[**Creating the Model** 10](#_Toc64972429)

[**Creating our Entities** 13](#_Toc64972430)

[**Creating the relationships** 15](#_Toc64972431)

[**Creating and Configuring The Business Object** 17](#_Toc64972432)

[**Creating the UI** 21](#_Toc64972433)

[**Creating our UI Template** 21](#_Toc64972434)

[**Editing the Overview page of the Floor Plan Manager** 23](#_Toc64972435)

[**Configure the UIBB** 26](#_Toc64972436)

[**Creating a Role** 33](#_Toc64972437)

[**Configure the Role** 36](#_Toc64972438)

[**Creating and Configuring our Business Activities** 38](#_Toc64972439)

[**Creating our Change Requests** 42](#_Toc64972440)

[**Linking our UI and the Change Request header** 45](#_Toc64972441)

[**Connecting our two UI components.** 48](#_Toc64972442)

[**Build The workflow** 50](#_Toc64972443)

# **Creating a Custom Data Model**

# **Introduction**

In MDG the data model is a central part of the, MDG provides several domains specific and out of the box applications, which means it can be used from the start with little or no configuration. Although it is a common requirement for customers to adapt the MDG data models to suit their specific needs.

## **Basic Data Model for Material**

It is important to understand that MDG not only delivers the data fields in a model but as well comes with the standard business rules to check for completeness and consistency. You can either centralise the maintenance of processing controlling data on the MDG Hub by using the standard backend transactions for material maintenance or you can decentralise the maintenance process controlling data.

### **Storage Types**

You assign a storage and use type to specify whether and how the master data can be changed in Master Data Governance. The storage type and use type also indicated which database tables are generated by the system.

There are four types:

**- (Type 1) Changeable vis change Request; Generated Database Tables:**

The master data of this storage type can be changed in Master Data Governance with a change Request. The system generates all necessary database tables. The common key of these tables is the entity itself, the edition and the entity types that are assigned to the entity type through leading relationships

**- (Type 2) Changeable w/o change Request; Generated text tables:**

The master data of this storage and use type can be changed in Master Data Governance without a change request. The system generated only the check and text tables with the entity type as well as with the entity types assigned to the entity types through leading relationships as fixed key fields

***- (Type 3) Not Changeable via MDG; No Generated tables:***

The master data of this storage and use type cannot be changed in Master Data Governance, for this reason the system does not generate database tables. Instead the system derives the values from the domain that is assigned to the data element.

***- (Type 4) Changeable via Other Entity Type; Generated Database Tables:***

The Master data of this storage and use type can be changed in Master Data Governance only with a change request of an entity type with storage and use type1. The entity needs to be in a relationship with the relationship type leading and assigned as the To-entity to an entity type with storage and use type Type1. The system generates check tables as described for storage and use type1 but also generated the entity types that are assigned through the qualifying relationships as key fields.

### **Relationship type**

*Referencing* Specifies the From-Entity as an attribute of the To- Entity

*Leading* Specifies the From-Entity on a higher level than the To-Entity type. The From-Entity is automatically taking as the key in generated tales.

*Qualifying* Identical to Leading From-Entity is on a higher level than the To-Entity type. The From-Entity is automatically taking as the key in generated tales.

1: N: This cardinality represents a mandatory relationship in which one or more To- Entity types can be assigned to a From-Entity Type. This cardinality is valid for relationships Leading, Qualifying and Referencing. The from entity type is a required attribute in the Referencing relationship

0: N This cardinality represents an optional relationship in which any number of To- Entity types can be assigned to a From-Entity Type

## **Business Scenario**

You are an MDG Consultant working for a customer, the customer has a requirement for a custom data model they can use to keep a record of their employees and their current and previous addresses. You have to create this custom data model from scratch.

# **Step by Step Procedure**

## **Creating Custom Elements**

|  |  |
| --- | --- |
| You can use elements in the system but for our demo we want to build our own elements, so we have the freedom to customise their descriptions labels and everything.  Enter the transaction se11 and then type in the name of your element in our case YYEMPL then choose to create and choose data element. |  |
| As you can see we have entered the description of Employee ID. You can fill in the details as shown  Here we are leveraging the domain that is already in the system NUMC |  |
| Don’t forget to fill in a field label for your data element in our case this will be EmployeeID. |  |
| We now must make another element one for the employee age, so the name will be YYAGE. Like before click create and choose data element |  |
| Like before we are inputting our own description and we are again leveraging a domain in the system already num2, which is a 2-digit number. |  |
| Again we must fill in the field labels of our data element. Fill in as shown or create your own field labels |  |
| We need to create a data element for Address type. We will name this as YYADTYP… create the element as done before  We will first put in the description as we did before except this type we will not leverage a domain already in the system but create our own. To do this we will input our domain name YYADTYP. Then select enter.  You should get the message as shown. |  |
| Now to create our own we will double click on our domain element; we will be asked to save the changes to YYADTYP as shown select yes. You will then be asked if you want to create the domain again select yes |  |
| Fill-in the domain fields as shown In the picture  Setting the descriptio0n, the data type and the number of characters in that data type. |  |
| Now we also want to add in a value range this allows us to utilise a drop-down option for address type given the user certain values when using our data elements. Add in the values shown in the image below. Return to your data element and input the field label as we did before.  Short addtyp & medium address type  Now lastly we need to create an employee name.  You will also need to create a domain for this like previously although with no value range. Create as before  YYNAME – Employee Name |  |

## **Creating our Custom Data Model**

### **Creating the Model**

|  |  |
| --- | --- |
| Now we can start creating the model. The first thing we need to do is create a blank data model with no entities  We start off by entering transaction mdgimg.  Open general settings 🡪  Data Modelling 🡪  Edit data Model. |  |
| We are choosing to edit this z model as it is a blank data model with no entities meaning we can use it to create our data model from scratch.  We can simply rename the data model right here without clicking any buttons. |  |
| You can but there is no need to rename the data model, but just rename the description, in our case we have renamed the description to Employee Model Demo CK. |  |
| Double click the entity types.  Here we can see that there is no entities for this model.  We can start inputting our own entities with the elements we created earlier.  Choose new entries in the top left corner highlighted in the image shown. |  |

### **Creating our Entities**

|  |  |
| --- | --- |
| We must first start with a new entity, so I am calling this YYEMPLOYE  Next, we choose relationship Type 1. Essentially meaning the data can persist in a generated table but importantly it means change requests can be created on this entity, additionally it means we can change this element through change requests.  We also want to leverage the data element we created YYEMPL.  Deletion could be allowed, or it could not be allowed it is a preferential choice but in our case we will allow it.  One thing we must do is, put an Active area. In our case we need to put MDG. Now we need to save it |  |
| Now we must click on the attributes and essentially provide the employee with attributes ie. the elements we created Name and Age.  As shown in the picture  Now we save, our type 1 entity is now complete |  |
| We repeat these steps, creating a new entity. Return by choosing entity types and then click new entries.  We will be creating an entity for storing the address, this time we will name it YYADDRESS  In this case we will select Type 4 as the storage type. Type 4 you don’t need an element as it is tied to the attribute its built on. We will put in a description as shown. |  |
| Again we must provide our address entity with attributes,  In our case we will utilize the data elements already created in the system, as we haven’t made our own custom elements.  You can enter the details of the attributes as seen in the picture. |  |
| Lastly We will create an entity call YYADTYP.  We will choose type three storage which means there will be no table for this entity which makes sense as it is a key and doesn’t need a table  This entity will be used to bind the other two entities together in a relationship.  When saved you will now get 2 errors in relation to relationships once we save our attributes. That is because we have yet to define the relationships of our entities. |  |

### **Creating the relationships**

|  |  |
| --- | --- |
| Click into relationships in the left column.  Choose new entries. We will be building a relationship from Employee to Address we must also give our relationship a name in our as we will name it yylead. It is a leading relationship with a cardinality of 1 : N meaning one employee can have many addresses  Now we will save, once we save, we will get an error that is okay the error is essentially telling us we have not distinguished our key to bind the two tables together. |  |
| Like before chose new entries.  Now we need to create our 2nd relationship, From address type to the address. This relationsip will be Qualifying.  This relationship will bind the two entities Employee and Address together.  Again we will have the cardinality of 1 :N meaning one address type can have many addresses |  |
| Now we can save. With that step we nearly have created our model. Make sure you have activated your model. To do that click inactive data models in the left column choose our model ZC and then click activate. You will see at the bottom it says check and generate structures, highlighting it is generating the structure of our model.  Now we will have seen an error in relation to business object. So we have to create a business object to create that linkage |  |

### **Creating and Configuring The Business Object**

|  |  |
| --- | --- |
| To link our model we have to create a business object, at this stage all we need is a very simple Business Object with a description.  It has a lot of other functionality we will see in the future but before now we just need a simple Business Object type  Open transaction MDGIMG 🡪  Open General Settings 🡪  Data Modelling 🡪  Define Business Object Type Codes. |  |
| We want to create a simplified object so we will create one from scratch We will click new entries and then fill in the details of our business object as shown in the image  You can name your Object however you pleased and can further input the description as you please |  |
| Now we will return to the Edit Data Model screen.  We will then select our data model ZC and click entity types.  We then need to select our type one entity which is YYEMPLOYE. Select this then select Business object types from the column on the left highlighted in the picture |  |
| Choose new entries. Now we add in the business object we just created YEMP.  Tick the box to suggest it’s a root then click save. Then click save. Now we are almost complete creating our data model. |  |
| Now lastly we just need to Define the entity type to be used by business object type.  Return to the mdimg screen,  General settings 🡪  Data Modelling 🡪  Define Entity Type to be used by Business Object |  |
| Select new entries,  Input the details as shown, Business Object type will be the object we just created YEMP, our data model will be ZC which we created previously and lastly our entity type will be our employee entity YYEMPLOYE  Now we have completed the data model. Now we must create our UI. We don’t need to create SMT for now, we will require them when we are editing an already used data model although not for a newly created data model |  |

## **Creating the UI**

### **Creating our UI Template**

|  |  |
| --- | --- |
| Okay so let’s start off with manage UI Configurations, there are many ways to do this, colleagues have preferential choices as to whether they go through the object dictionary or not.  We chose UI configuration as it is a lot more straightforward. |  |
| So, the template we are going to use is USMD\_OVP\_GEN.  It stands for: USMD overview page Generated Templates  This is very similar to MM UI.  We will first select this and then click copy. |  |
| The first thing we do is change the affixes.  Affixes means it comes by default with ‘Z\_’ highlighted in the image shown.  We will change this to reflect our data model. To do this select change affixes, and input ZC instead of the Z |  |
| We can now also remove ’GEN\_TEMPLATE’ from the first line and remove ‘TEMPLATE’ from the third line just so our UI can align with the rest of the naming conventions of the other UI’s.  Your UI should now appear as shown in the image shown in the image.  Now lastly, we have to choose Start Deep copy highlighted.  You should see a notification at the bottom saying the deep copy is complete. |  |

### **Editing the Overview page of the Floor Plan Manager**

|  |  |
| --- | --- |
| Now we should click on the ZC\_USMD\_OVP, this is where the relationship between the model will be built.  First thing we do is write the business object we created in the field USMD\_OTC.    In our case that business object is YEMP. We are just inputting the business object although it is important to note the other fields, such as CRTYPE and ACTION. It highlights that there can be many fields tied to the UI making it very flexible and very specific to a process or a process type.  Choose save. |  |
| Now what we want to do is edit the overview page of the floorplan so the ZC\_USMD\_GEN\_OVP. We can open this by returning to the menu and selecting the tab as shown. |  |
| Select edit, and the first thing we will do is change the page ID in the left column from MAIN to ZCEMPLOYEE. |  |
| Next, we will select the arrow on UIBB highlighted in the image to open the drop-down menu and select ‘form component’. |  |
| We will now enter our configuration ID, again we will write a name containing our model to make it recognisable.  ‘ZC\_EMPLOYEE’  We also have to fill in title, this is the description of our form, so lets name this Employee Form and choose save. |  |

### **Configure the UIBB**

|  |  |
| --- | --- |
| Now we have to configure the UIBB, the form now exists that we have saved it. Although there is still no configuration so we have to configure the UIBB.  Click ‘configure UIBB’ and then We will just name this description as an employee form, and select ok. |  |
| Now this in the important factor, the model that we have built, how does this component connect with our model, well it does so through the feeder class.  Okay so search for the feeder class we will use. ‘CL\_mdg\_bs\_guibb\_form’. Then click enter and it should automatically select the feeder class we have just entered.  Now we have to do the connection between that feeder class and our model. In the component field enter Z\* and search. As shown. |  |
| Now as you can remember our model is ZC , so we have to look out for that. As you can see there are various different connectors, such as multi record processing, hierarchy processing and single processing. We want to choose the single processing as we will be adding just one employee in our case. Select the result ZSP\_ZC |  |
| Now we must select the object of the component, select the search box in the field Object Name. We wil see the entities we built and how the system is making that connection, Select ‘YYEMPLOYE’. |  |
| Select ok all the content of that entity should now appear, this is how the UI is trying to make some sort of connection with the model. Now you will see any fields here that we didn’t originally create, these are created automatically in the backend. We can hide these |  |
| To do this scroll down to the fields in the section below. Select all the fields except leave the entities we created unselected such as age, name and employee ID. Then click the bin icon to delete them all. |  |
| Now the UI is just the entities we created and no backend fields or buttons, although it looks a bit messy so we will have to change this  Now the easiest way to move these into different positions is through drag and drop although, it can be a little awkward and can get messy. A more straightforward way is to use the grid. When you select one of your labels you can see on the right side there is a section called position, this corresponds to the Y-axis and X-axis grid. |  |
| Now you can rearrange them in any order, but in our case, we will rearrange them in ID, Name and then Age like shown.  We can then save this, and exit. |  |
| In this case we aren’t making a form component but making a list component. Now we want to do this process all over again, for the table. So we will return to our ZC\_USMD\_OVP\_GEN screen. |  |
| Now like before we will fill out the details of our element  Configuration ID is ZC\_ADDRESS\_LIST and we also must put in a description in the title field we will put in Addresses.  Click save then like before click configure UIBB. |  |
| We can use the description of Address list here save as local object like before. Now we must choose the feeder class again like we did before.  This time we are going to use the feeder class CL\_MDG\_BS\_GUIBB\_LIST enter this and then select enter |  |
| Like before we must input our component and our corresponding object. Like before we will choose the single processing component ZSP\_ZC. Except this time we will choose a different object as we are building the UI for our address list we will naturally choose the object YYaddress as shown. Then select ok |  |
| Now we will see a template table with our fields. We will also see a series of buttons above our table that we don’t need. We can simply remove these like we did in our previous UI config for employees. To do this click on tool bar schema and then select all, although one of these buttons we need is create so unselect the button that highlights new then click the bin to delete that tool bar. As shown |  |
| Now going back to the tab List UIBB Schema, we can see we have some description fields for address type and two for country key we can go ahead select and delete the 2nd country key description and the description for address type like usual by clicking the bin icon.  Now we can save and it should appear as shown in the image |  |
| Now we can return to the OVP\_ZC\_USMD\_GEN\_OVP screen.  Frist thing we need to do is Change the main page id to match the configuration id of our employee Form. So in our case change the id to ZC\_Employee. As shown in the picture  And then save this |  |

### **Creating a Role**

|  |  |
| --- | --- |
| So now we have made the two forms, for the employee and for the address list although we have yet to make the connection with the MDG change request so this will be our next task.  To do this return to Easy Access menu and enter the transaction ‘pfcg’. Now we will have to make a roll so the we can assign it to our user and the UI is visible in NWBC.  In the search bar type in the name of the roll we will create so for eg: YCK\_MDGX\_MENU\_05. Then select the icon labelled single role to create this role |  |
| Enter the description as shown and then choose save |  |
| Okay what we want to do is inherit the role of a SAP role already created to do this we select the box derive role under transaction inheritance and search for the role ‘ sap\_mdgx\_menu\_05’  Select enter and save. This will change our menu and add the roles and will additionally add applications |  |
| Now we must change the authorisations of this role to ensure we can edit it correctly, to do this open the authorizations tab and click Change Authorization Data |  |
| Select the pencil icon beside change Authorization Data, click the tick on the pop up screen, then select all authorizations and save and return to our role. Choose to save.  Now we must add our user to do this enter the user tab and the input your user id in this case it is I352778 as shown then save. |  |

### **Configure the Role**

|  |  |
| --- | --- |
| Now we must do some changes to this role to ensure it can process the UI we just created.  Open the menu and then open change request then open single object processing. |  |
| Right click on this and select details. Now this is tied to the UI configuration ZW now we need to tie this our configuration ZC  Select the search and then select your ui from the drop down screen. As shown then select the green tick. |  |
| Now double click on the single object processing before and click execute. This will show you what your UI will look like once executed in nwbc.  Although there is still no change request ties to this UI. |  |
| Now what we need to do is edit the search object, so like before right click on search object and select details. we want to select our model to make a connection. Input the ZC as shown, we can then click the tick and save our role and close the role configuration |  |

### **Creating and Configuring our Business Activities**

|  |  |
| --- | --- |
| Now we must create a business activity create business activity. |  |
| Now we need to create a business activity for both create and change, firstly what we will do is copy a sample activity and change them to suit our activity. Since it’s a custom object it is easier for us to select the Bus Activity SF1 & SF2 which are for create and change airlines. Scroll to sf1 and select both those rows as shown and select copy as in the top left corner highlighted in the picture given |  |
| Now we must edit the business activities to suit our model and needs. We first will change the name of the business activities now we can name these whatever we like in my case I am going to name them ZCCR for create and ZCCH for change. We also must input our descriptions and we must change the BO type to the business object we created so YEMP |  |
| Now we have connected the model with the business activity with the business object, but we have yet to connect the UI with all of them. So now lets add the UI to it |  |
| Now like before we will copy one and adapt it. Select the same business activity as shown In the picture then click copy as highlighted a the top |  |
| Now we only must edit three columns here, The business object type, the target UI Configuration and the business activity at the end  The other columns are okay to be left as is. We can fill in the business object type we created, the target UI configuration we created as shown and the business activity for create we just created ZCCR. Then click enter and save. |  |
| Now going back to the list, we need to select two views with Logical action change and create, as shown in the screen. Select these two rows then click copy as |  |
| Like before we will change the same three rows, Business object type to reflect the BO we created, Target UI configuration to reflect our Target UI configuration and the Business activity to reflect the two activities we created ZCCR and ZCCH. Just like shown in the image. Ensure we input the correct corresponding activates. Click enter and save |  |

### **Creating our Change Requests**

|  |  |
| --- | --- |
| Now we still must add the change request type link to our UI. So, what we will do is create our own change request type and link it to our UI configuration. Again, we wont create one from scratch but simple copy another and adapt it. In this case we will copy the create material change request.  Find MAT01 select it and then click copy as |  |
| Now we have to change the type of change request we can name this whatever we like but in my case I am going to continue with the same naming conventions including my Model in the name so it is easily distinguishable. Next change the model from MM to ZC , change our descirptio0n to create employee ZC and lastly change the main entity type to our Entity YYEmployee . No need to change the workflow as we want to use the same workflow as the create material. Click enter and copy all |  |
| Now on our new screen select the Change request we just created and click entity types and delete all those entity types are they are coming from the MAT02 we don’t need these. Then select new and input our entity as shown in the image. |  |
| Then go to the business activities. Now as you can see in the business activity it is MAT01 , we again have to delete this as this business activity is unrelated to our model and UI. Then Click new entries and add the business activity ZCCR. As shown in the picture . Click enter and save |  |
| Now that we have created a change request for creating an employee, we also must create a change request for changing one. Returning to the list of change request, select the change request we just created and click copy as. |  |
| Now we have to change the type of change request from ZCEMPCRE to a name of our chosen I am choosing ZCEMPCHG. We don’t have to change anything else click save and click copy all. |  |
| Now we must select the change request we just created ZCEMPCHG and click on business activities. Now we remove the ZCCR business activity and click new entries and input our ZCCH activity as shown in the picture then click save. Now we have finished creating the change requests. |  |

### **Linking our UI and the Change Request header**

|  |  |
| --- | --- |
| The last thing we must do is create a link to the change request, right now the UI hasn’t been linked with the change request. We need to go back into manage UI configurations screen. Scroll down to our model and as you can see it is a circle when we hover over that it will say ‘configurations Missing’. |  |
| Click on details and we will open the governance communicator. We can name the description the same as the name so ZC\_USMD\_OVP. On the new screen select settings then click new and crWires. This connects it to the UI |  |
| Next we have to fill in the details of the crWires, first be input our page id, in our case this is ZC\_EMPLOYEE. Now we have to change the source component from FPM\_FORM\_UIBB to the component of our model. Which is FPM\_FORM\_UIBB\_GL2. Lastly we need to add in our source config name which is also ZC\_EMPLOYEE. Now we can save this |  |
| No we also need to add the UIBB to the configuration, like before select settings click new and choose SearchUIBB’s. |  |
|  |  |

### **Connecting our two UI components.**

|  |  |
| --- | --- |
| Now we need to form a relationship between our UI screens. To do this return to our manage UI configurations screen. Scroll down to our ZC created model and click on the UI Configuration link ZC\_USMD\_GEN\_OVP. |  |
| On the new screen click wire scheme, we will use the wire scheme to create a relationship between the two UI components. Then click ‘+ wire’ in the top left corner to create a new wire. You should now have a screen as shown |  |
| On the right hand side we need to fill in the details of this wire. Frstly we need to fill in component ‘FPM\_FORM\_UIBB\_GL2’  This will then automatically fill in the config id. Next we have to fill in the connector class, here we can search and choose from the list. We will choose BOL Identity Mapping this is essentially allowing the component to identify and connect with itself. |  |
| It should appear as shown in the picture. Next we need to create a 2nd Wire, again like before click onn the + Wire and begin to fill out its details. |  |
| Here we need to connect the list to the source so in our case connect the address to the component. We then have to insert the connector type, in our case it is a lead selection, and input a standard port identified. Lastly, we have to fill out the connector class like before this time we are choosing Bol relation connector for BP. Your details should appear as shown in the picture. Now we can save. |  |

### **Build The workflow**

|  |  |
| --- | --- |
| We are going to create a workflow for our Model now.  We are going to create a rule based workflow.  Double click on  ‘Configure Rule Based Workflow’  On the new screen put in your data model ZX. |  |
| Locate the decision table in the left column and open the Non-User Agent Table. |  |
| You can fill out the table as shown in the picture |  |
| Next we will fil out the Single Value decision table.  You can fill it out as shown in the picture.  We can leave the User Agent table blank.  Make sure to save and activate. |  |

