



PUBLIC

How-To Get Default Values in UIs for Single Object Maintenance based on FPM BOL Feeder Classes

Applicable Releases:

From MDG 7.0 onwards and from SAP S/4HANA 1511 onwards

Version 2.1

November 2023

Document History

Document Version	Description
1.0	First official release of this guide (November 2014)
2.0	Minor updates & layout (November 2020)
2.1	Changed Layout (November 2023)

1	BUSINESS SCENARIO.....	4
2	DEFAULT VALUES IN THE CHANGE REQUEST UIBB	4
2.1	By Code – Using enhancements	4
2.2	By Code – Using Inheritance and WebDynpro Customizing	6
2.3	By WebDynpro Personalization	8
2.4	By Business Add-In (BAI) – not possible.....	8
3	DEFAULT VALUES FOR THE ENTITY	8
3.1	By Code	8
3.2	By WebDynpro Personalization	10
3.3	By Business Add-In (BAI)	11
4	ADDITIONAL INFORMATION	12
4.1	Further Reading	12
4.1.1	Information on SAP MDG on SAP S/4HANA	12
4.1.2	SAP Roadmap Explorer	12
4.1.3	Related Information	12
4.2	SAP Notes.....	12

1 Business Scenario

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.

This How-To Guide describes the solution to initialize some fields of the Single Object Maintenance UI with default values.

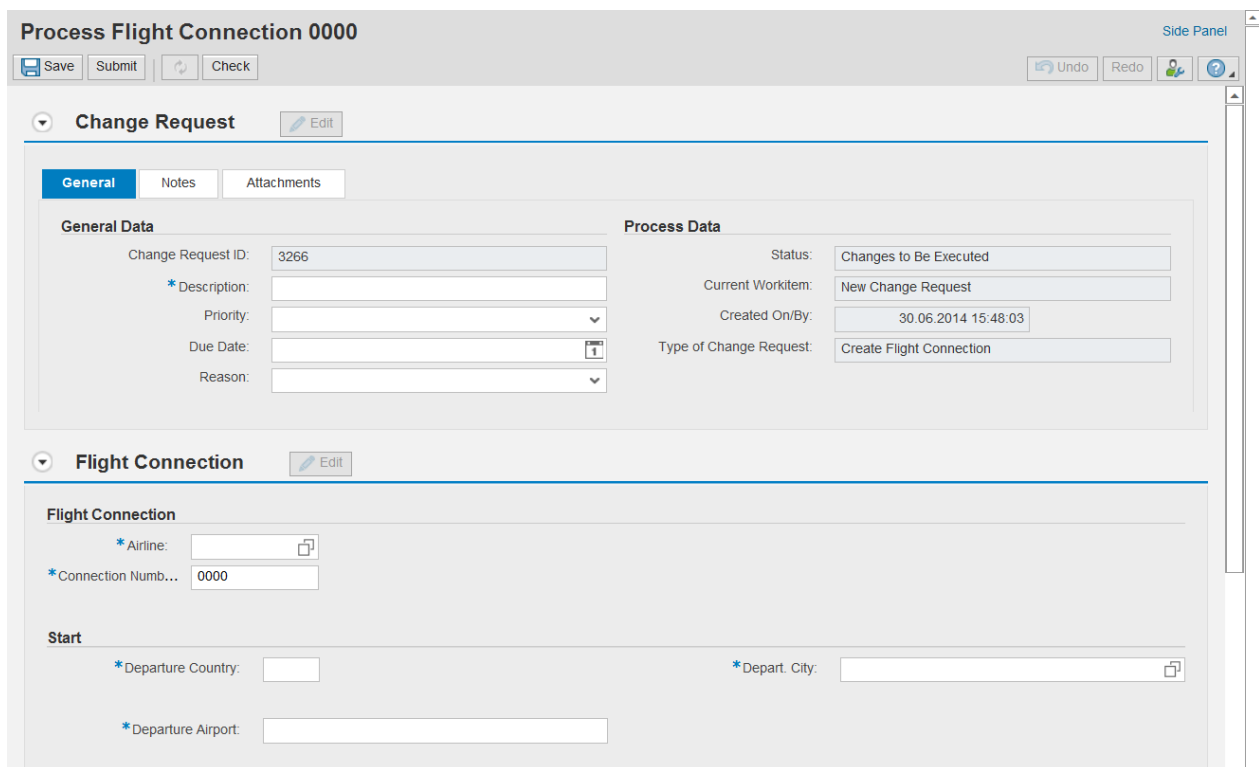


Figure 1

In Figure 1 you see the UI for data model SF, entity type PFLI which we will use to demonstrate different ways of setting default values. The upper UI Building Block (UIBB) for the change request data is not part of the floorplan definition – at least not at design time. It will be added to the floorplan at runtime.

The lower UIBB belongs to the floorplan of the currently running application. The UI and its feeder classes can be SAP-owned or customer-owned.

2 Default values in the change request UIBB

2.1 By Code – Using enhancements

Solving a task by writing own code always provides the highest flexibility. In the case of the Change Request UIBB, the feeder classes are provided by SAP, and you should not modify this code. The best place to add your own code is the end of method `IF_FPM_GUIBB_FORM~PROCESS_EVENT` in class

CL_USMD_CR_GUIBB_GENERAL_DATA. To do so you can create an implementation of the implicit enhancement point at the end of the method. The change request is represented by the object **mo_entity**. By using method **get_properties** you get a structure which already contains some data like the change request ID and type.

```

Class Builder: Class CL_USMD_CR_GUIBB_GENERAL_DATA Display
Method IF_FPM_GUIBB_FORM-PROCESS_EVENT Active
71
72 *****"SE:(1) Class CL_USMD_CR
73 *$*-Start: (1)-----*$*
74 ENHANCEMENT 1 Z2_CR_HEADER_DEFAULT_VALUES. "active version
75 IF mo_entity is bound.
76
77 mo_entity->if_bol_bo_property_access-get_properties(
78 IMPORTING
79 es_attributes = ls_cr_attributes ).
80
81 IF ls_cr_attributes-text is initial and ls_cr_attributes-type = 'SFP01'.
82 mo_entity->if_bol_bo_property_access-set_property(
83 exporting
84 iv_attr_name = 'TEXT'
85 iv_value = 'New Connections with A380' ).
86
87 mo_entity->if_bol_bo_property_access-set_property(
88 exporting
89 iv_attr_name = 'PRIORITY'
90 iv_value = '02' ).
91
92 mo_entity->if_bol_bo_property_access-set_property(
93 exporting
94 iv_attr_name = 'DUE_DATE'
95 iv_value = '20141025' ).
96
97 ENDIF.
98 endif.
99 ENDENHANCEMENT.
100 *$*-End: (1)-----*$*
101 ENDMETHOD.

```

Figure 2 - Implementation of the implicit Enhancement Point 'End of Method'

General Data		Process Data	
Change Request ID:	3275	Status:	Changes to Be Executed
*Description:	New Connections with A380	Current Workitem:	New Change Request
Priority:	High	Created On/By:	30.06.2014 18:01:05
Due Date:	25.10.2014	Changed On/By:	30.06.2014 18:01:33
Reason:		Type of Change Request:	Create Flight Connection

Figure 3 - The Change Request UIBB directly after starting the application with the enhancement

Figure 2 shows that you only need just a few lines of code to set default values for the header data of a change request. However, since you have to enhance a method at the end, you might prefer class enhancements over source code enhancements. In this case you would create a Post Exit for method **IF_FPM_GUIBB_FORM~PROCESS_EVENT** with access to private and protected components of **CL_USMD_CR_GUIBB_GENERAL_DATA**.

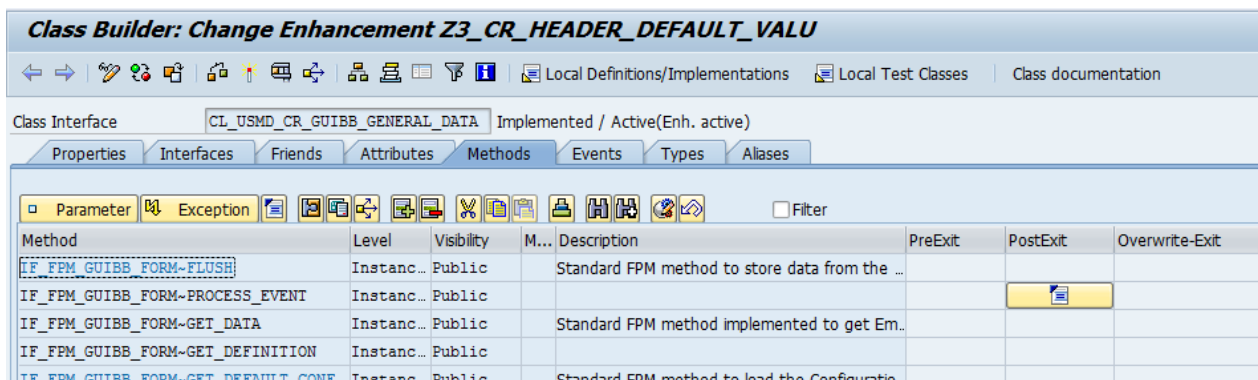


Figure 4 - Methods tab with existing class enhancements

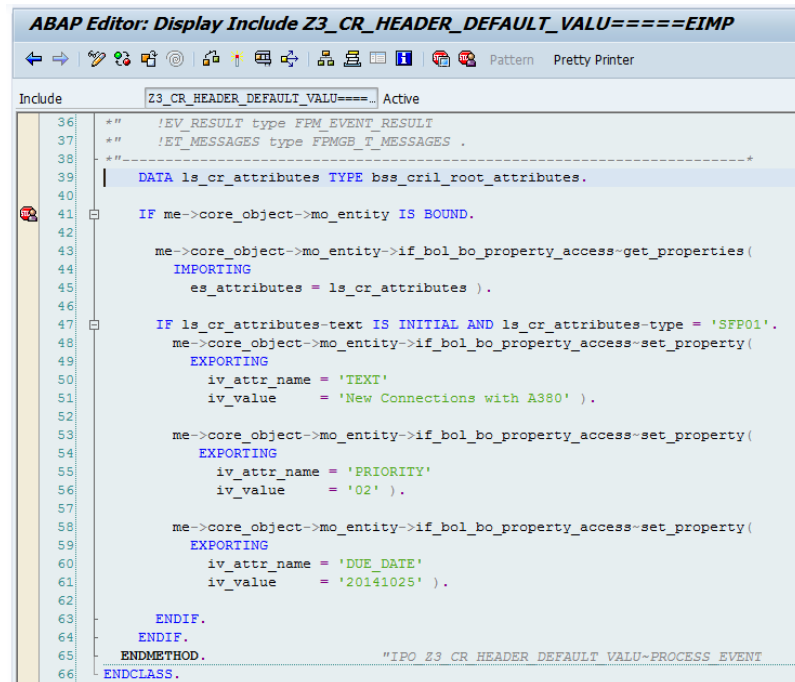


Figure 5 - The same code as in Figure 2 but this time as a class enhancement

2.2 By Code – Using Inheritance and WebDynpro Customizing

Instead of enhancing the existing feeder class, it is also possible to replace it with an own class. This class inherits from the original feeder class and will only redefine relevant methods, in this case **PROCESS_EVENT**. The only limitation of this procedure is that the original feeder class must not be final.

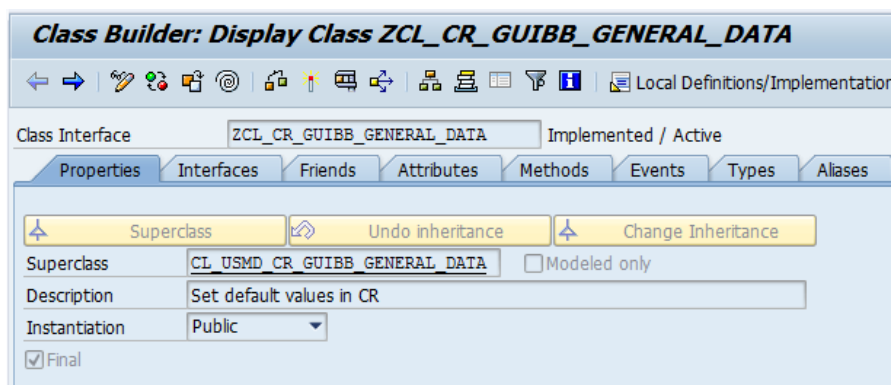


Figure 6 - Create a subclass of the feeder class

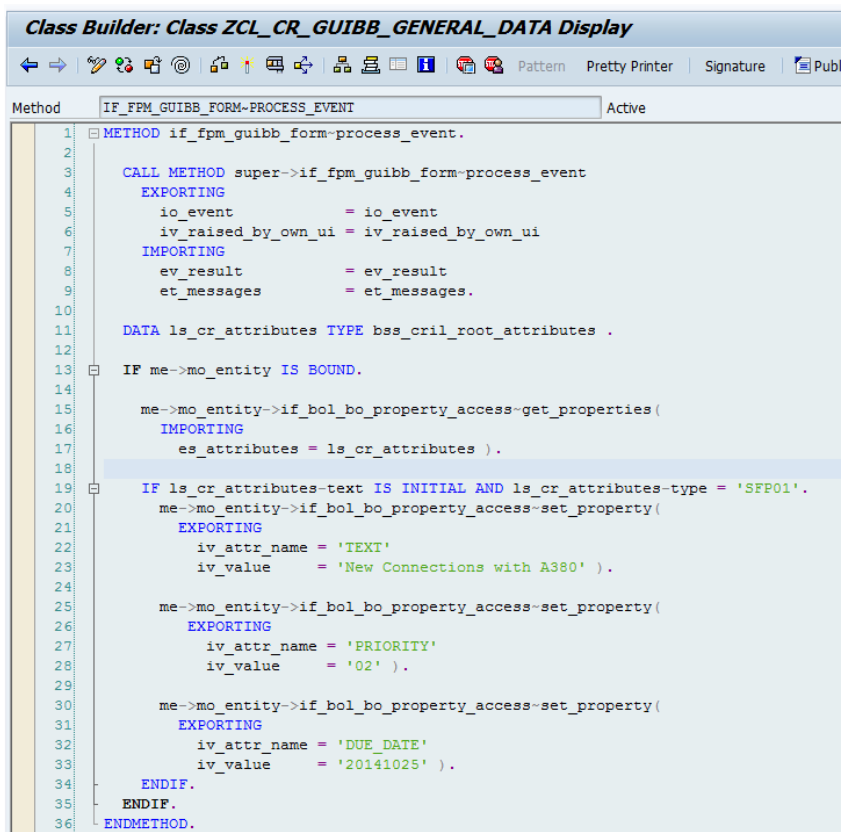


Figure 7 - The code of the redefined method. Please pay attention to the call of the super class

The next task is the link from the UI-configuration to the new feeder class. This can be done by creating a customizing for the UIBB-configuration. This customizing must be of the same name as the configuration. In the customizing, you can enter a feeder class which overwrites the class from the configuration level.

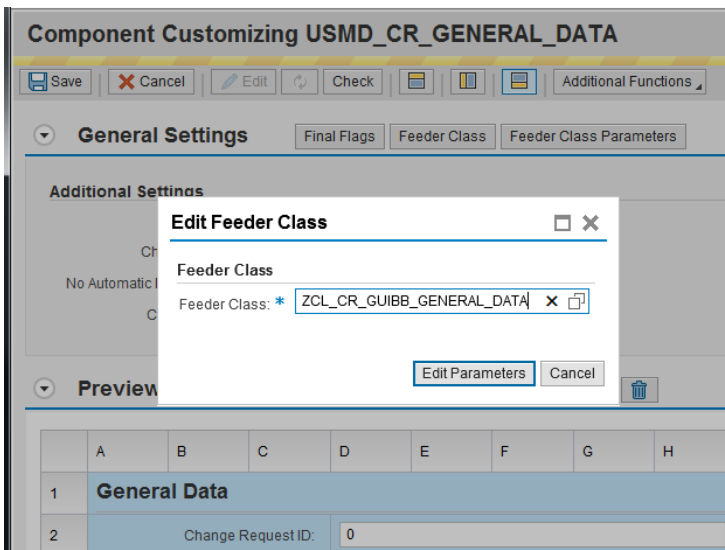


Figure 8 - Exchange the feeder class by running the configurator in customizing mode

Replacing the feeder class via customizing provides you with a modification-free possibility to link your own code with an existing UI. However, customizing means that this solution is client dependent.

2.3 By WebDynpro Personalization

The Web Dynpro personalization is available via the context menu of UI elements. Setting default values is supported by input fields and dropdown boxes (only DropDownByKey). However, dropdown boxes don't accept default values if an event handler is registered on the **ON_SELECT** event. Since this is the case for the fields *Priority* and *Reason*, you can only populate the input fields of change request UIBB.

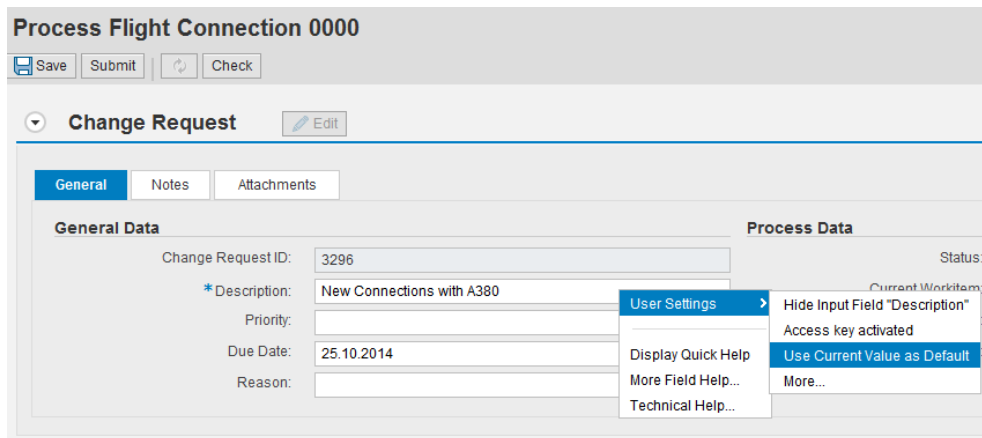


Figure 9 - The context menu of an input field

This solution works at user level, which makes it very flexible for the individual user. In contrast to the solution with enhancements, you don't need a development user here. The default values take effect immediately since you set them in the system in which you work in. Transporting these settings is not necessary and not possible. Default values can't be set when the application runs in administration mode (URL-parameter `sap-config-mode=X`). In other words, an administrator can't set default values for all users.

Personalization is only possible if it is allowed at system level and not disabled at application level.

Deleting the default value can be done the same way as it was set:

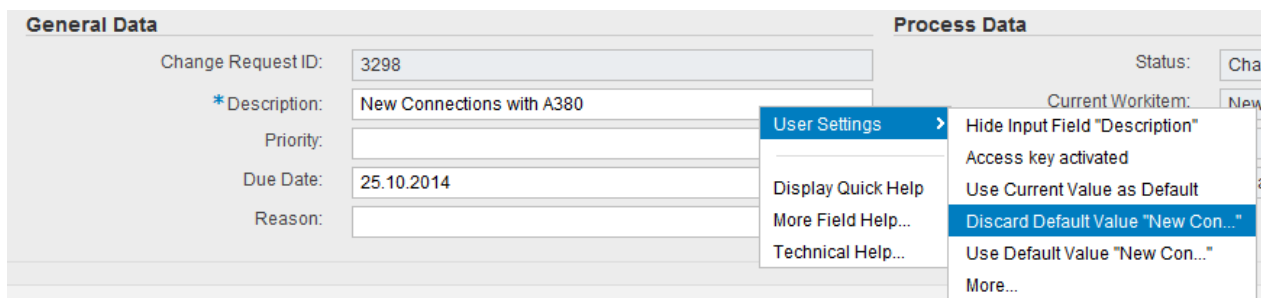


Figure 10 - The context menu of an input field with a default value

2.4 By Business Add-In (BAdI) – not possible

There is no BAdI which allows you to set default values in the change request header.

3 Default values for the entity

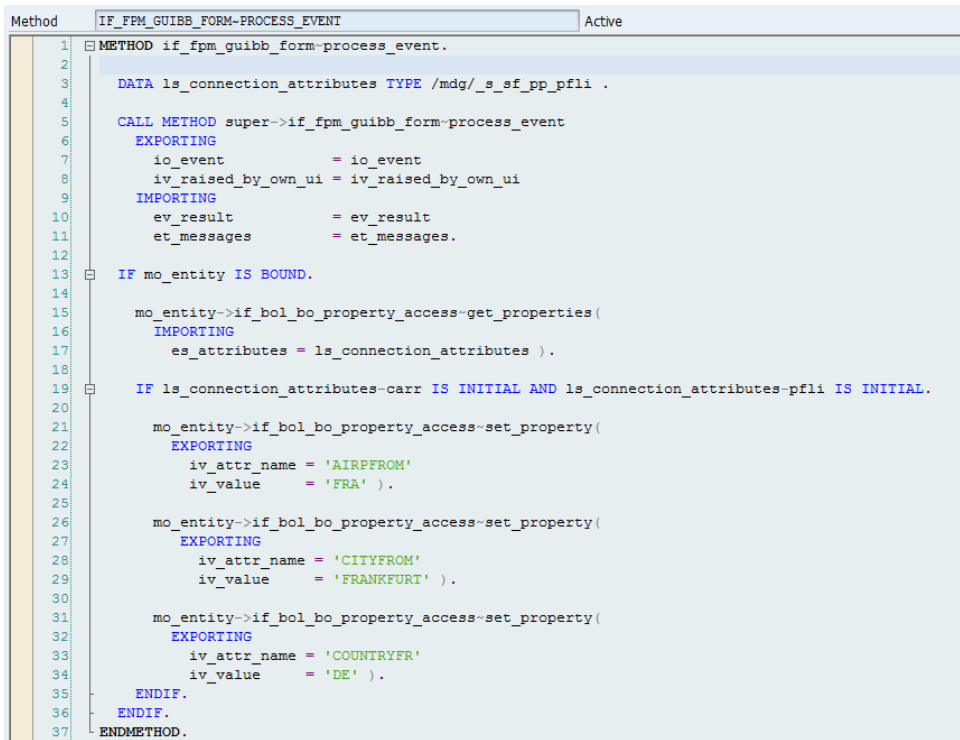
3.1 By Code

Setting default values in the UIBB of an entity works very similar to setting values in the UIBB of the change request. There are several possibilities for where to place the code depending on which feeder classes are used. If you use class `CL_MDG_BS_GUIBB_FORM` directly as a feeder class, you can either enhance this

class or inherit from it and redefine method `IF_FPM_GUIBB_FORM~PROCESS_EVENT`. The latter one is the better solution, and you should create a subclass of `CL_MDG_BS_GUIBB_FORM` if your feeder class does not already inherit from this class. In this case, a programming error only impacts the UIBB using this feeder class instead of all form UIBBs.

If you are dealing with an SAP-owned UIBB, you can introduce your own feeder class by using Web Dynpro customizing, as already described.

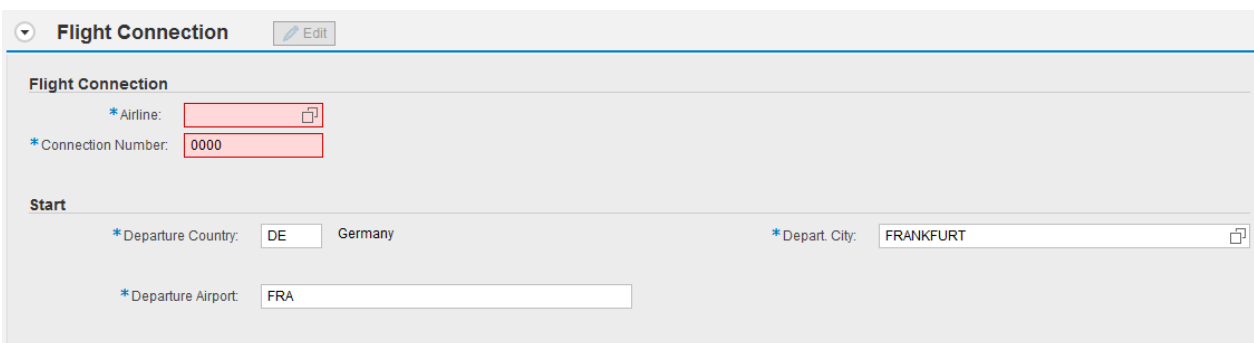
The following two screenshots show the example code of a feeder class which inherits from `CL_MDG_BS_GUIBB_FORM` and the relevant part of the UIBB for the creation of flight connections.



```

Method: IF_FPM_GUIBB_FORM~PROCESS_EVENT | Active
1  METHOD if_fpm_guibb_form-process_event.
2
3  DATA ls_connection_attributes TYPE /mdg/_s_sf_pp_pfli .
4
5  CALL METHOD super->if_fpm_guibb_form-process_event
6  EXPORTING
7    io_event          = io_event
8    iv_raised_by_own_ui = iv_raised_by_own_ui
9  IMPORTING
10   ev_result          = ev_result
11   et_messages        = et_messages.
12
13  IF mo_entity IS BOUND.
14
15    mo_entity->if_boi_bo_property_access-get_properties(
16      IMPORTING
17        es_attributes = ls_connection_attributes ).
18
19    IF ls_connection_attributes-carr IS INITIAL AND ls_connection_attributes-pfli IS INITIAL.
20
21      mo_entity->if_boi_bo_property_access-set_property(
22        EXPORTING
23          iv_attr_name = 'AIRPFROM'
24          iv_value     = 'FRA' ).
25
26      mo_entity->if_boi_bo_property_access-set_property(
27        EXPORTING
28          iv_attr_name = 'CITYFROM'
29          iv_value     = 'FRANKFURT' ).
30
31      mo_entity->if_boi_bo_property_access-set_property(
32        EXPORTING
33          iv_attr_name = 'COUNTRYFR'
34          iv_value     = 'DE' ).
35    ENDIF.
36  ENDIF.
37  ENDMETHOD.
  
```

Figure 11 - A redefinition of `PROCESS_EVENT` with the code for default values



Flight Connection [Edit]

* Airline:

* Connection Number:

Start

* Departure Country: Germany

* Depart. City:

* Departure Airport:

Figure 12 - The result of the code of the previous figure

In this case, we have set values for an entity which does not yet have a key, which is why the upper two fields are highlighted in red. After populating these fields, the highlighting and the error message in the message area will disappear.

3.2 By WebDynpro Personalization

In general, you can use the Web Dynpro personalization for entity data the same way as for the change request data. Nevertheless, there is one small difference you should remember. The change request UIBB is always in change mode when it appears for the first time, and changes are possible. In contrast, there are scenarios when entity data are in display mode when the UIBB is displayed the first time. This is the case when you click the link with the entity name in the search results.

Result List: 26 records found

Pe...	Airline	Connection Number	Description (long ...	Rank	Departure airport
	LH	0400		100,00	FRA
	LH	0401		100,00	JFK
	LH	0402		100,00	FRA
	LH	0404		100,00	AIY
	LH	0454		100,00	FRA
	LH	0455		100,00	SFO
	LH	1453		100,00	ASP
	LH	1706		100,00	JFK

Figure 13 - A search result list

After clicking the link '0404', the entity will be displayed. All entity data are in input fields which are in read-only mode. The user has a default value for the unit of measure of the distance, but it will not be set by Web Dynpro because the field is read-only.

Process Flight Connection

Connection Details

Flight Connection

Airline: LH
Connection Number: 0404

Departure

Airport: AIY
Country: US

Destination

Airport: JKT
Country: ID

Details

Distance: 1.000,0000
Flight time: 9:00
Charter flt: ☐

After choosing the *Edit* button, the input fields of non-key fields change to edit mode. Nevertheless, the field for the unit of measure remains blank because the fields are technically the same as before, and Web Dynpro only applies default values when a field is displayed for the first time.

Connection Details

Flight Connection

Airline:
Connection Number:

Departure

* Airport:
* Country:

Destination

* Airport:
* Country:

Details

Distance:
Flight time:
Charter flt: ☐

Figure 14 - Switching from display to edit mode does not set default values from personalization

This example was only shown for the sake of completeness. It does not represent a big limitation because default values are more important during the creation of an entity, a process when all input fields are open for changes immediately.

3.3 By Business Add-In (BAdI)

BAdI **USMD_RULE_SERVICE**, method **DERIVE_ENTITY** is not primarily made for default values, but it can be used to do so. It is called after filling in the key fields of the entity and triggering a roundtrip (e.g., by pressing ENTER). It is not possible to set the key fields of the entity. The filter values of this BAdI are entity type and model.

BAdI **USMD_UI_EVENT2**, which you might know from the previous UI for object maintenance (application **USMD_ENTITY_VALUE2**), is not called by the new UI because you have now the possibility to inherit from the generic feeder classes provided by SAP.

www.sap.com

4 Additional Information

4.1 Further Reading

4.1.1 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](#)

4.1.2 SAP Roadmap Explorer

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

4.1.3 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](#)

4.2 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

