





GUIDE FOR FLOORPLAN MANAGER

FS-PM 5.1



Icons used in this Guide

Icon	Meaning
	Caution
	Example
	Note
	Recommendation

Abbreviations used in this Guide

BTX:	Business Transaction
FPM:	Floor Plan Manager
BTS:	Business Transaction Scheduler
UIBB:	FPM user interface building block

Table of Contents

GETTING STARTED	4
1.1 ABOUT THIS DOCUMENT	4
1.2 IMPORTANT NOTES	4
1.3 RELATED INFORMATION/FURTHER USEFUL LINKS	4
GENERAL COMMENTS ON HOW TO IMPLEMENT A FPM UI.....	5
IMPLEMENTATION OF AN FPM UI – OVERVIEW OF MAJOR STEPS.....	6
IMPLEMENTATION OF A BUSINESS TRANSACTION – THE STEPS IN DETAIL	6

Getting Started

1.1 About this Document

This document contains the information **customers and partners** need to implement floor plan manager user interfaces for business process Change of the solution SAP Policy Management (FS-PM) 5.1 .

This document is aimed to serve as a guide and give assistance when you face the task of implementing a new user interface. It describes in more or less detail all steps that are needed for this. For complete technical details of the Floor Plan Manager tool check separate documentation of this tool.

The description in this document is based on SAP Policy Management (FS-PM) 5.1. It is written for developers in partner organizations or on customer site. It is assumed that the reader of this document already has good skills in FS-PM development and is familiar with Floor Plan Manager (FPM) concepts.

1.2 Important Notes



Read the following documents on last-minute changes:

- SAP Notes for additional information on findings discovered after release to customer. SAP Notes may contain further links to main Notes for each release.
- *ReleaseNotes.doc* for additional information that might not be included in this document.

1.3 Related Information/Further Useful Links

For more information that is not covered in this guide, see below:

- [SAP HELP PORTAL \(http://help.sap.com/\)](http://help.sap.com/)
 - SAP NetWeaver (information about Workbench, BAdIs, etc.)
 - SAP NetWeaver (information about Floorplan Manager for Web Dynpro ABAP)
 - FS-PM itself
 - Business Partner
 - Organizational Management
 - External systems such as

- SAP Collections & Disbursements (FS-CD)
 - SAP Claims Management (FS-CM)
 - SAP Commissions Management (ICM)
 - SAP Reinsurance Management (FS-RI)
- And so on.
- [Industry Solution Master Guide - SAP for Insurance \(](https://sapneth3.wdf.sap.corp/~form/sapnet? SHORTKEY=00200797470000066278&)
 - SAP Policy Management: Enhancement Guide
 - Further information provided in the SAP for Insurance - Solution Manager Content
- Online documentation
- SAP Notes
- And so on.

General Comments on How to Implement a FPM UI

The description in this guide refers to the shipped FPM UI for business transaction “Change Payment Frequency” of business process Change as a sample. This business transaction has been chosen as a “representative” business transaction as it comprehends all common functions of a business transaction while not being too complex in detail.

For this business transaction SAP delivers the FPM feeder class /PM0/CL_ABX_GUIBB_FEEDER and default BadI implementation /PM0/CL_ABX_FPM_GET_DATA_PMD, as well as service classes /PM0/CL_ABX_GUIBB_BTRANS_DB and /PM0/CL_ABX_REMOTEDISC_SRVC. The feeder class is meant to be generic to all business transactions and any business transaction specific screen manipulation and data dependent logic are contained in BadI methods. In addition, the BadI contains a method for customer specific checks to be carried out prior to the execution of the background BTX. A different BadI implementation will be required for each business transaction. The service classes handle data base accesses and the execution of the background business transaction and apply to all business transactions.

In addition, three FPM components are delivered to implement the UI. /PM0/FORM_UIBB_P_B_S_PMD_AMD contains input screen fields for policy, effective date etc. as well as display fields to help identify the correct policy. This form can be reused for other business transactions. /PM0/LIST_UIBB_P_B_S_PMD_AMD is a list of information about the contracts of the policy along with dropdown boxes used to change the information. This UIBB is specific to a business transaction. /PM0/FORM_OIF_P_B_S_PMD_AMD is the component that packages the FORM and LIST UIBB and provides the command buttons. The control of the activation of the commands is contained in the BadI implementation.

The data fields that are available for use on a FPM screen are controlled by customizing on a business transaction basis. Adding a field to this customizing makes it accessible in FPM UIBB design and the data container used to execute the background business transaction. Please note that there are some fields defined in the

/PM0/FORM_UIBB_P_B_S_PMD_AMD that are not displayed on the screen. These are for internal use in the feeder class to transfer data between the FORM and LIST and should not be deleted from the FORM.

Implementation of an FPM UI – overview of major steps

In this chapter we outline the major steps that are needed to implement your business transaction.



SAP recommends to basically following the steps as listed in the sequence below.

- Execute the business transaction in dialog mode to determine what data is required. Also notice which fields are display only.
- Determine the layout of the fields to be used in the contract list UIBB.
- Check if all of the fields you require are in customizing. If not, add them.
- Execute the FPM Configurator for the list UIBB. Use /PM0/CL_ABX_GUIBB_FEEDER as the feeder class. Select the fields in the proper order for the list. Make any input field for changed values a dropdown box.
- Execute the FPM Configurator for the OIF Component. Select PM0/FORM_UIBB_P_B_S_PMD_AMD as the first UIBB and the list UIBB that you created as the second UIBB. Add command buttons for Save, Cancel, Load Policy and Calculate. Use /PM0/FORM_OIF_P_B_S_PMD_AMD as a guide.
- Create an application configuration using the OIF Component you created in the previous step. Leave the IDR component blank. Then create a corresponding web Dynpro application.
- Create a Badi implementation for /PM0/ABP_FPM_GET_DATA_BADI. This is a filtered Badi, so use the technical name of the business transaction as the filter. For example, change payment frequency is P_B_S_PMD_AMD.
- Code the Badi methods GET_DATA and GET_DATA_LIST with any logic specific to the business transaction. Use /PM0/CL_ABP_FPM_GET_DATA_PMD as a guide. Perform any command button actions in GET_DATA.

Implementation of a business transaction – the steps in detail

This chapter, as already mentioned, refers to FPM UI for Change Payment Frequency as sample. We will give a description which objects have had to be implemented in which way to get this representative business transaction up and running.

1. Execute the business transaction in dialog mode to determine what data is required. Also notice which fields are display only.

Policy Process Goto Actions System Help

Execute

Change Contract Journal

Selection

Policy Number	51000000000009878
Effective Date	01.01.2009
Appl. Receipt Date	31.12.2008
Application Date	31.12.2008
Application Number	
Processing Level	32767
Change Reason	2 Customer Request
Suppl. Change Reason	Change the payment frequency

BTrans Filter

Policy

Policy Start	01.01.2007	00:00:00	CET Central Europe
Sales Product Name	Family Coverage		

Policyholder

Business Partner	200263	Martin Braun Test
Date of Birth	16.08.1981	
Gender Name	male	
Address	<Standardadresse> Musterstr. 22 / 88400 Biberach TEST	

Application Process Goto Actions System Help

Change: Change Payment Frequency

Complete Charges

FS-PM Navigation Tree

- Appl.
 - Family Coverage
 - PH0001 Personal Liability
 - EH0010 Personal Liability
 - DH0007_FinancialLoss
 - DH0006 Key Risk
 - DH0003 Bodily Inj. Prop. Dam
 - DH0002 Rental Property Damag
 - DH0001_Construction Owner's F
 - EH0011 Children of Legal Age
 - PH0001 Personal Liability
 - PH0002 Livestock Owner Liability
 - PS0001 Household

- Change Option
- Overview
 - Application Journal
 - Contract Journal
 - Geplante Bearbeitungen
 - Transaction/Date
 - Business Partner
 - Policy
 - Transfer
 - Vertragserneuerungshistorie
 - Correspondence History
 - Collections/Disbursements
 - Commission
 - Benefit Cases/Claims
 - Reinsurance

Policy Number: 5100000000009878 Effective Date: 01.01.2008

Application Number: 133231 BP Mode: Application

Policyholder: Martin Braun Test Status: In Process

Dunning Status: No Dunning Paid to:

Payment Frequency

Payment Frequency: 01 Annually

Premium

According to Pymt Freq. Proportional Annually

fr. To

Tariff Prem. Sum Ins			104,00
Coll. Premium B. Tax	104,00	0,00	104,00
Cumul. Tax Chrg	19,76		
Coll. Prem. Aft. Tax	123,76	0,00	123,76

Premium Taxes

Cty	Currency	Tax Amo...	Tax Rate	Tax Type	Name of Tax Type	Tax Scope	ScpPrerm
DE	EUR	19,76	19,000	5	Insurance Tax		3 Subject t

Coverage

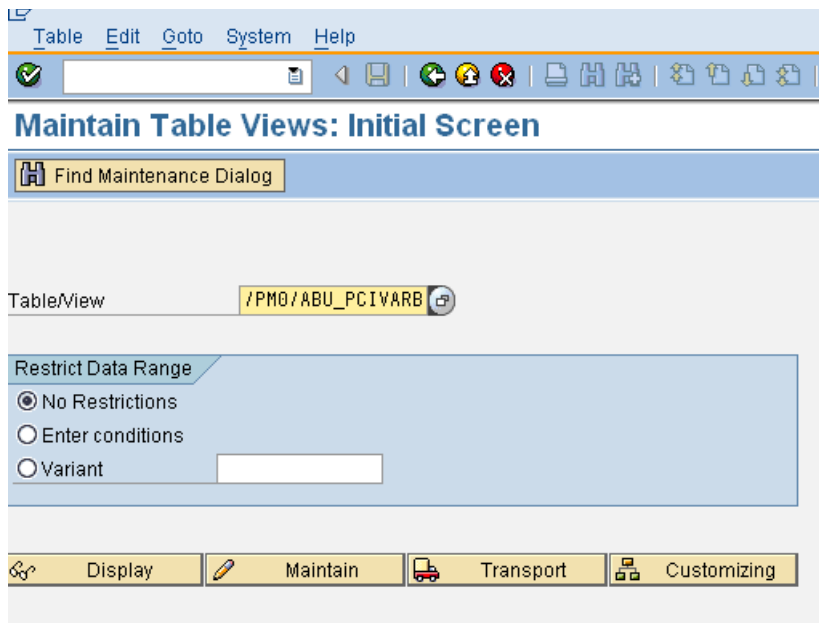
Coverage Number	Elementary Product Name	Insured Object Name
1	EH0010 Personal Liability	

2. Decide which fields you want to use. Design the layout of the columns. (This is a manual activity.)

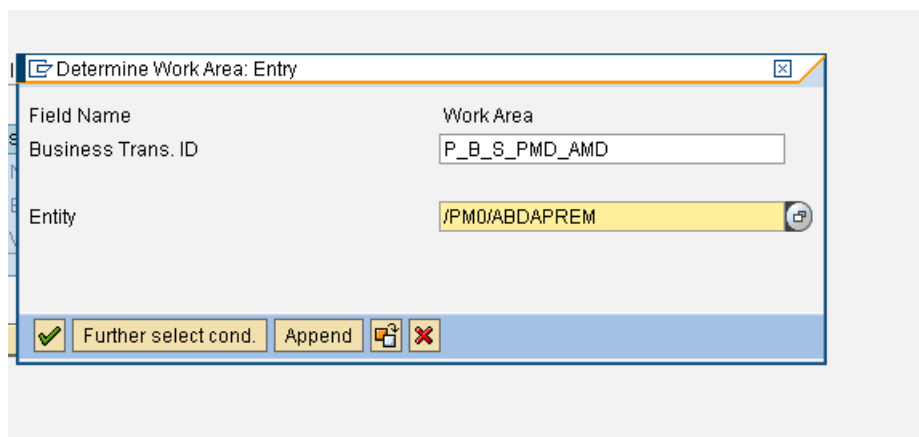
Contract Number	Product	Previous Payment Frequency	Previous Premium After Tax	Del.AnnlPrem.Afrt Tx	New Payment Frequency	New Premium After Tax	Del.AnnlPrem.Afrt Tx
-----------------	---------	----------------------------	----------------------------	----------------------	-----------------------	-----------------------	----------------------

3. Check if all of the fields you require are in customizing. If not, add them.

Run transaction SM30.



Select "Maintain" and enter the business transaction ID and the entity required. In this case the premium entity is chosen.



Hit enter.

Table View Edit Goto Selection Utilities System Help

Change View "Configuration of BTrans.-Specific Comparison List": Overv

New Entries

Bus. Trans. ID: P_B_S_PMD_AMD
Entity: /PM0/ABDAPREM

Configuration of BTrans.-Specific Comparison List

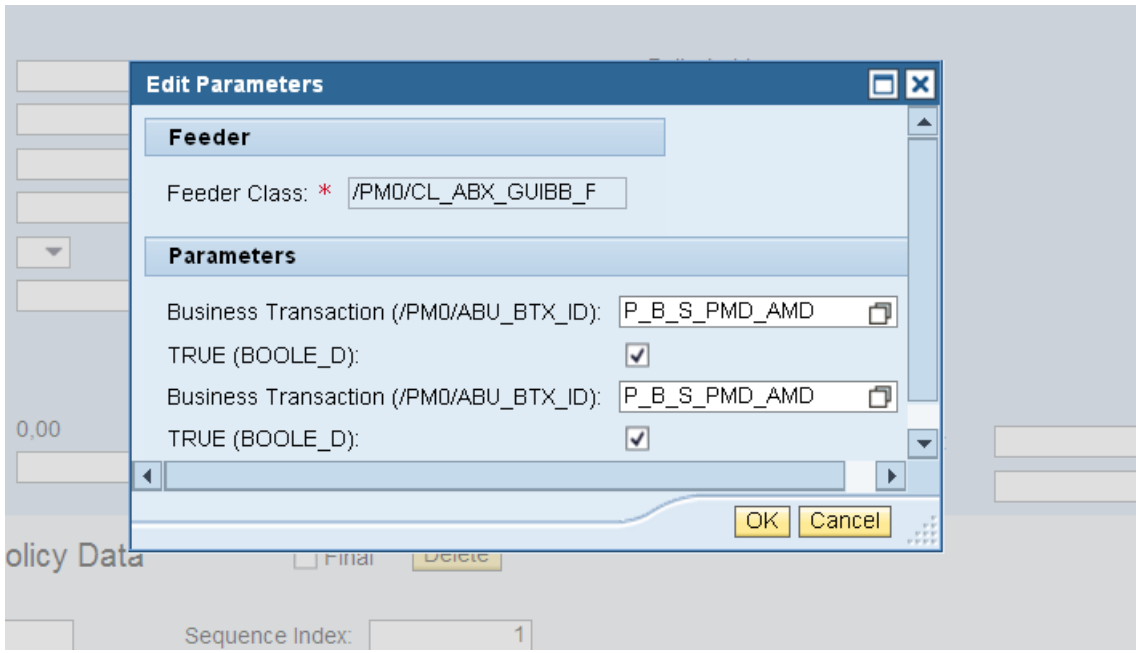
Field Name	Entity Description	Description of Attribute Properties	Excl.Attr.
ANNPREMAFTTAX_AM	Basis Premium	Annual Premium After Taxes	<input type="checkbox"/>
CURRENCY_ID	Basis Premium	Currency	<input type="checkbox"/>
PAYFRQ_CD	Basis Premium	Payment Frequency	<input type="checkbox"/>
PAYFRQ_TT	Basis Premium	Payment Frequency	<input type="checkbox"/>
POLICYPRODUCT_ID	Basis Premium	Contract Key	<input type="checkbox"/>
PREMAFTERTAX_AM	Basis Premium	Premium After Tax	<input type="checkbox"/>
PREMIUM_ID	Basis Premium	Premium Key	<input type="checkbox"/>

If the required field is not listed, enter it by selecting New Entries.

- Execute the FPM Configurator for the list UIBB. Use /PM0/CL_ABX_GUIBB_FEEDER as the feeder class. Select the fields in the proper order for the list. Make any input field for changed values a dropdown box. When creating either a form or list UIBB there are four parameters required. The first and third are for the business transaction and the second and fourth must be set to true.



The Feeder class processes the command buttons that are delivered with the Payment Frequency Change BTX. If additional command buttons are required, a copy of the /PM0/CL_ABX_GUIBB_FEEDER feeder class should be made and logic added for the additional commands. If additional commands are needed, consideration should be given to adding them to the LIST UIBB if possible. In this way, the delivered feeder class could be reused.



Component Configuration /PM0/LIST_UIBB_P_B_S_PMD_AMD

Save Check New Window Deep-Copy Display View: Component-Defined

Edit Feeder Edit Parameters Configure Toolbar Configure Columns

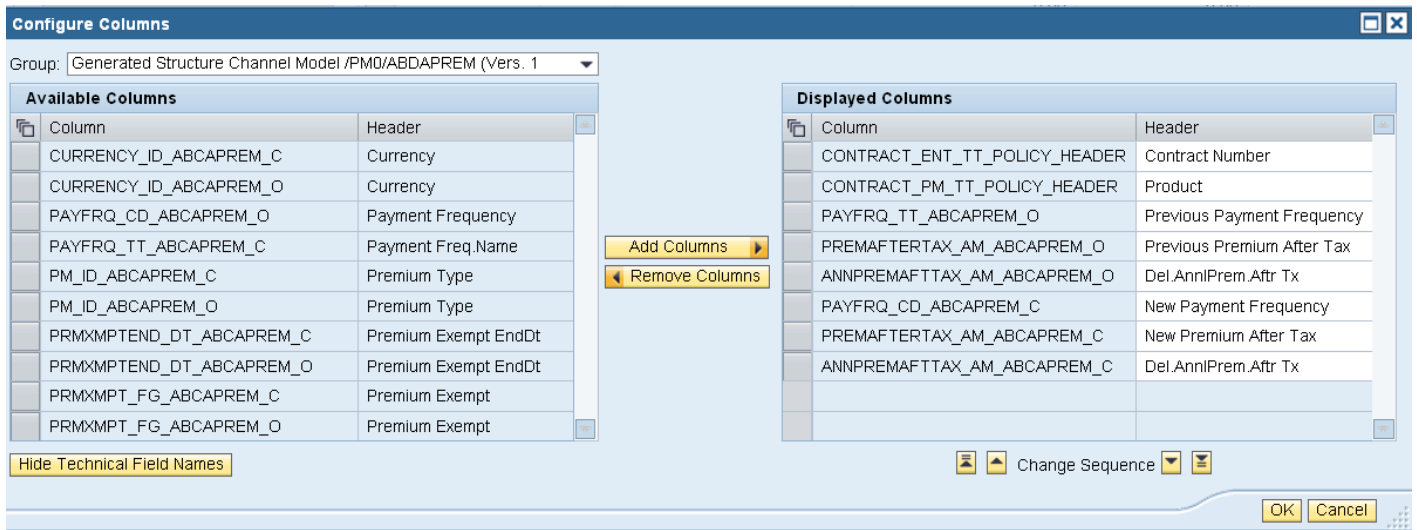
Contract Number	Product	Previous Payment Frequency	Previous Premium After Tax
			0,00
			0,00
			0,00
			0,00

Attributes List Final

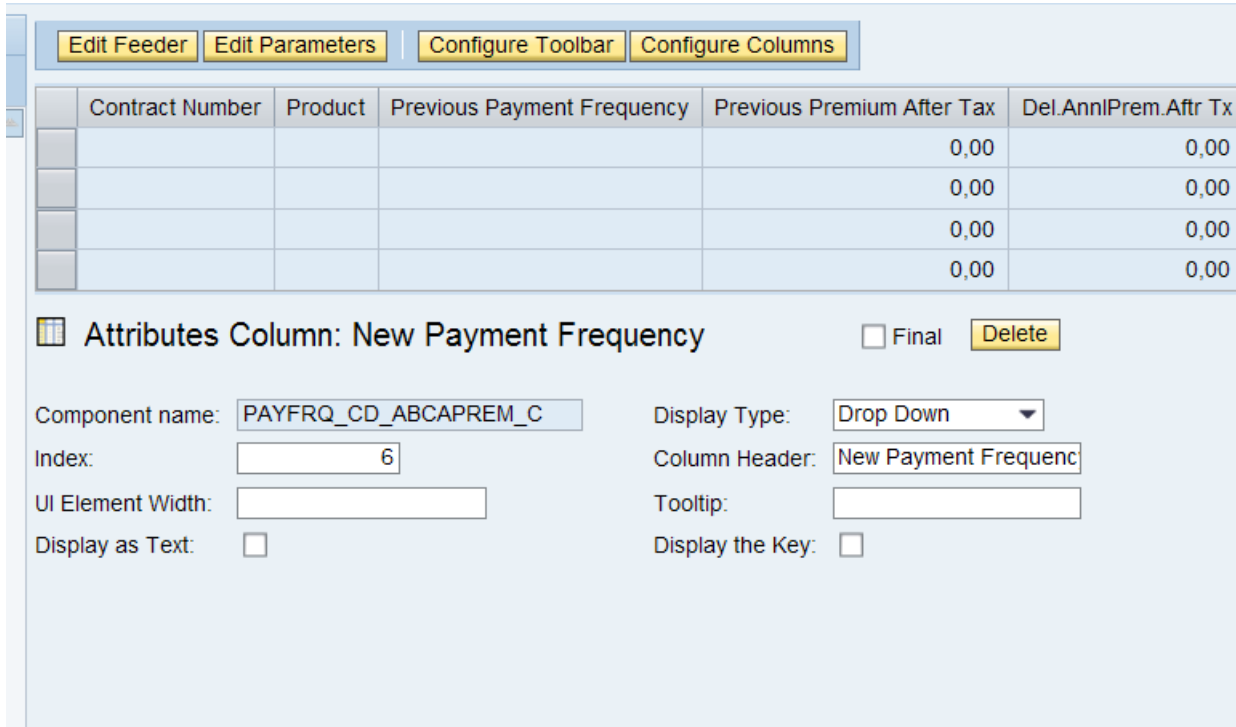
Feeder Class: /PM0/CL_ABX_GUIBB_FEEDER Text:

Lead Selection Action Assignment:

Hierarchy
 Expand All Collapse All
 Element
 List
 - Toolbar
 - Column: Contract Number
 - Column: Product
 - Column: Previous Paym...
 - Column: Previous Premi...
 - Column: Del.AnnIPrem...
 - Column: New Payment F...
 - Column: New Premium A...
 - Column: Del.AnnIPrem...
 - Settings



Since the New Payment Frequency field is an input to the business transaction, make it a drop down field.



- Execute the FPM Configurator for the OIF Component. Select /PM0/FORM_UIBB_P_B_S_PMD_AMD as the first UIBB and the list UIBB that you created as the second UIBB. Add command buttons for Save, Cancel, Load Policy and Calculate. Use /PM0/FORM_OIF_P_B_S_PMD_AMD as a guide.

Application title Help

Form UIBB (1)

Component Name: FPM_FORM_UIBB
View: FORM_WINDOW
Configuration Name: /PM0/FORM_UIBB_P_B_S_PMD_AMD

List UIBB (2)

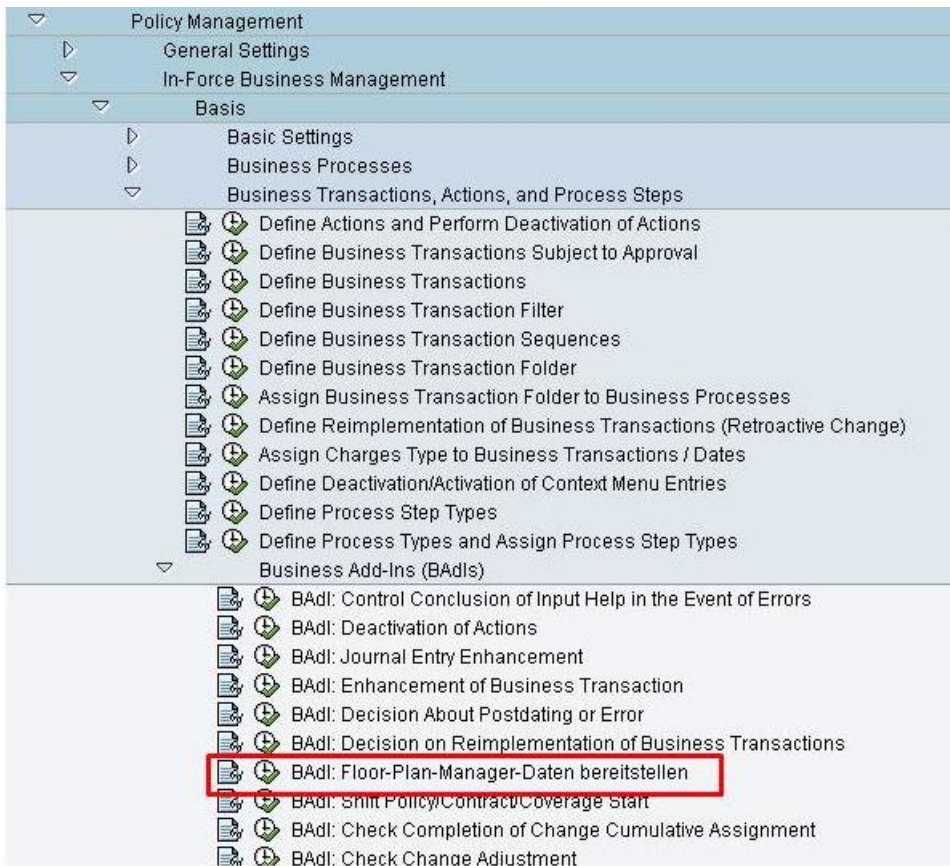
Component Name: FPM_LIST_UIBB
View: LIST_WINDOW
Configuration Name: /PM0/LIST_UIBB_P_B_S_PMD_AMD

6. Create an application configuration using the OIF Component you created in the previous step. Leave the IDR component blank. Then create a corresponding web Dynpro application.

Assignment of Component Configurations

Component Usage	Component	Implementation	Configuration
<ul style="list-style-type: none"> ▼ /PM0/FORM_P_B_S_PMD_AMD ▪ IDR_USAGE 	FPM_OIF_COMPONENT	FPM_OIF_COMPONENT	/PM0/FORM_OIF_P_B_S_PMD_AMD
	FPM_IDR_COMPONENT	FPM_IDR_COMPONENT	

7. Create a BadI implementation for /PM0/ABP_FPM_GET_DATA_BADI. This is a filtered BadI, so use the technical name of the business transaction as the filter. For example, change payment frequency is P_B_S_PMD_AMD. Code the BadI methods GET_DATA and GET_DATA_LIST with any logic specific to the business transaction. Use /PM0/CL_ABP_FPM_GET_DATA_PMD as a guide. Perform any command button actions in GET_DATA.



Method: /PM0/IF_FPM_GET_BTX_DATA~GET_DATA	This method is used to access and change data that is currently on the screen. It can also be used to change the attributes of screen or list fields. In addition, the enabling or disabling of command buttons is possible. The method is called after every FPM event and after the GET_DATA method of the FPM feeder class has successfully executed.
IMPORTING PARAMETERS	
IO_EVENTID	A reference to the FPM event that is currently being processed. This allows the execution of specialized logic depending on the event.
IO_CNR_OIF	A reference to the current instance of the OIF component. This allows control of the components in the screen including the command buttons.
CHANGING PARAMETERS	

CS_DATA	The current form data.
CT_FIELD_USAGE	The attributes of the current form data.
CT_ACTION_USAGE	The attributes of the defined FPM events.
EXPORTING PARAMETERS	
EV_DATA_CHANGED	A flag indicating that the form data has changed.
EV_FIELD_USAGE_CHANGED	A flag indicating that the form data attributes have changed.
EV_ACTION_USAGE_CHANGED	A flag indicating that the FPM attributes have changed.

Method: /PM0/IF_FPM_GET_BT_X_DATA~GET_DATA_LIST	This method is used to change or check data in the list. It can also be used to change the attributes of the list columns and report errors back to FPM.
IMPORTING PARAMETERS	
IR_EVENTID	A reference to the FPM event that is currently being processed. This allows the execution of specialized logic depending on the event.
IV_LOAD_ERROR	A flag indicating if there was some error loading or checking the policy.
CHANGING PARAMETERS	
CT_DATA	A table of the current values of the list. Each row in the table corresponds to a row in the list.
CT_FIELD_USAGE	The attributes of the columns of the list.
CT_MESSAGES	A table containing messages to be displayed after the current event.

Method: /PM0/IF_FPM_GET_BT_X_DATA~CHECK	This method is used to implement any specific checks not covered by the standard Change Badi checks. (/pm0/abp_btx_badi). The parameters are identical to the parameters of the 'is_inactive' method of that Badi.
IMPORTING PARAMETERS	
IS_KEY	UBOI for /PM0/ABDACOV

IS_EXECUTION_KEY	Key for Execution Level of Business Transactions
IS_ACCESS_EXEC	Checks Whether Bus. Trans. Can Be Executed
IT_EXCEPTION	Exception Table
IV_EXECUTION_LEVEL	Execution Level of a Business Transaction
RETURNING PARAMETERS	
RF_INACTIVE	True if checks fail, false if checks succeed.