

# Kofax Process Director

## Advanced Configuration Guide

Version: 7.8

Date: 2020-06-17

The KOFAX logo is displayed in a bold, blue, sans-serif font. The letters are thick and closely spaced, with a consistent weight throughout the word.

# Legal Notice

© 2019 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

# Table of Contents

About this guide.....	6
Work with projects.....	7
What is a project?.....	7
Create a project.....	7
For SAP GUI.....	7
For the Process Director Web Application.....	7
Copy and compare projects.....	8
/EBY/PDBO_CONFIG_DUMP program settings.....	8
Configure an umbrella system.....	12
What is an umbrella system?.....	12
Configure an umbrella system.....	13
Configure remote function modules.....	13
Map umbrella workflow descriptions.....	14
Use centralized umbrella workflows.....	15
Map user IDs.....	15
Customize the SAP GUI detail screen.....	16
Configure an SAP GUI static tab.....	16
Configure the SAP GUI background screen.....	16
Configure the SAP GUI basic screen.....	17
Customize menus and buttons.....	18
Customize Web Application buttons.....	18
Available actions for Web Application settings.....	18
Examples.....	19
Customize SAP GUI buttons.....	20
Examples.....	21
Add a standard button.....	21
Hide a button.....	22
Develop BAdIs.....	23
What is a BAdI?.....	23
User Exits/BAdIs.....	23
Create a BAdI implementation.....	24
Best practice guidelines for BAdIs.....	25
Create custom rules.....	28
About custom rules.....	28

Create a custom preset.....	30
Create a custom check.....	31
Create a custom determination.....	32
Create the new determination.....	32
Create a new determination search.....	33
Define a document status.....	34
Define document status/substatus.....	34
Define icons for document status/substatus.....	34
Map message number to substatus code.....	35
Configure a popup.....	36
Change the popup title.....	36
Configure fields.....	36
Add values to dropdown fields.....	37
Activate subset posting.....	38
Create a custom process type.....	39
Create a new object type.....	39
Add a model level.....	42
Delete an object type.....	43
Display document item data in header.....	43
Configure the Generic Object Template (XY).....	44
Set up the BO data virtualization.....	47
Configure the jump to the Process Director document.....	48
Configure the Worklist.....	49
Nested set virtualization.....	49
Field selection.....	50
Settings.....	51
Create new object type.....	51
Add model level.....	52
Configure quantity conversion rules.....	54
What are quantity conversion rules?.....	54
Configure a quantity conversion rule.....	54
Integrate with SAP Business Workflow.....	55
About SAP Business Workflow integration.....	55
Create a new workflow definition.....	55
Create a container element.....	55
Set up the workflow start event.....	56
Add a container operation.....	57
Add an activity for the Process Director workflow step.....	57

Activate the workflow definition.....58

# About this guide

This guide is intended for implementation consultants who are installing and configuring the SAP/ABAP component of Process Director.

The guide describes advanced configuration tasks, such as developing BAdIs and creating custom rules.

For further information, refer to the following guides:

- *Process Director Import and Upgrade Guide*
- *Process Director Configuration Guide*
- *Process Director Reference Guide*
- *Process Director Troubleshooting Guide*

# Work with projects

## What is a project?

Projects allow you to test your configurations in a sandbox, without interfering with the production configuration. Kofax Process Director stores the entire project configuration separately from the standard configuration, so that changes to a project configuration do not affect the standard configuration and vice versa.

**Important** Projects are intended for use in internal systems only. You cannot use projects in production clients.

You can copy a project by downloading the project configuration to an XML file and then uploading it to another project. You can also compare two project configurations, irrespective of whether these are stored in the system or in XML files.

## Create a project

### For SAP GUI

To create a project, complete the following steps.

1. In the expert IMG, click **Change system settings** > **Project** > **Define projects** (/EBY/PDBO\_VPRJC) and add a new project for your configuration tests.
2. Select the SAP GUI menu item **System** > **User Profile** > **Own Data** and then click the **Parameters** tab.
3. Add the parameter ID /EBY/PDBO\_PROJECT to the table and set it to the project ID you wish to use.

**Important** The project ID is case-sensitive.

4. Click the **Save**  button.

### For the Process Director Web Application

Normally, project configurations are not available in the Process Director Web Application. However, if necessary, you can use this workaround:

1. Create a new RFC user.
2. Add the project parameter ID to the user profile of this RFC user and set the parameter to your project ID.
3. In the saplogon.properties file, configure a new connection to the SAP system and specify the user name and password of your RFC user in this connection. For example:

```

MYPROJECT.activate = yes
MYPROJECT.client = 800
MYPROJECT.user = MYRFC
MYPROJECT.passwd = myrfcuser
MYPROJECT.lang = EN
MYPROJECT.ashost = s47.r3.ebydos.local
MYPROJECT.sysnr = 0
MYPROJECT.maxconnections = 10

```

## Copy and compare projects

To copy a project, complete the following steps.

1. Go to transaction [/EBY/PDBO\\_CONF\\_DUMP](#).
2. In the **Task to perform** area, select whether you want download a configuration to an XML file, upload from an XML file, or compare one configuration with another.
3. The **Set of tables** option enables you to define which tables should be processed (customizing and system tables) during upload, download and comparison.

**Important** Uploading a Worklist configuration to a project in which the Worklist has already been configured is not recommended.

4. Enter the required [parameters](#) in the appropriate section and click the **Execute**  button.

**Note** After uploading a project configuration, you must generate the Worklist.

## /EBY/PDBO\_CONFIG\_DUMP program settings

Use the `/EBY/PDBO_CONFIG_DUMP` program to download the entire configuration of a Process Director project and store it in an XML file. The saved XML file can be uploaded to the same system or to other systems. You can also compare two project configurations, irrespective of whether these are stored in the system or in XML files. The program can also be run with transaction `/EBY/PDBO_CONF_DUMP`.

### Task to perform

Select whether you want download a configuration to an XML file, upload from an XML file, or compare one configuration with another.

In the Set of tables field you can specify which tables should be processed (customizing and system tables) during upload, download and comparison.

Variant	Description
CUS&EPD_FLDCAT	Exclude the field catalog from comparison
CUS&EPD_MENU	Exclude the menu entries from comparison

Variant	Description
CUS&EPD_WRKLIST	Exclude the Worklist entries from download
CUS&IPD_WRKLIST	Download only the Worklist entries
CUS&I_COCKPIT	Download only Kofax Process Director Accounts Payable tables

If you activate the upload option **Ignore wrong table type**, which allows upload of tables with a delivery class other than C or S, or tables that have delivery class C or S but no CLIENT field, you must enter these tables in the **Special tables** field. This feature should be used with care.

#### Download XML file

Setting	Description
To file	Name of the XML file in which the configuration will be stored.
Of document type	The document type to be downloaded. It includes independent tables of all the document types. When a table is document dependent: <ul style="list-style-type: none"> <li>• It contains all the entries where OBJ is empty.</li> <li>• It contains all the entries where OBJ is equal to that specified on the selection screen.</li> </ul>
From project	Project containing the configuration that will be stored.
Include system tables	Activate this option to include not only customized tables but also standard Process Director system tables in the downloaded XML file. This can be useful, for example, to check whether data in system tables has changed, or whether there are differences in configuration between two systems. Use the <a href="#">Compare</a> function to compare system tables.

#### Upload XML file

Only data from customized tables is uploaded. If system tables were included when the configuration was downloaded to XML, the data from these tables will not be uploaded.

**Note** Uploading a Worklist configuration to a project in which the Worklist has already been configured is not recommended.

Setting	Description
Request/Task	Name of a customizing request. All uploaded data is included in this request, which can then be released and transported to another system.
From file	Name of the XML file from which the configuration will be uploaded.
To project	Name of the project to which the configuration will be uploaded.

Setting	Description
Remove before upload	<p>Activate this option to remove existing entries in any PD or PDAP customizing table stored in the uploaded file.</p> <p>As it is possible to selectively download and upload parts of the configuration, only entries in those tables that are available in the XML file are removed before the upload; other tables remain untouched. You can also download and upload Process Director Accounts Payable tables.</p> <p>If you deactivate this option:</p> <ul style="list-style-type: none"> <li>• Entries that are present in the system database, but not in the XML file, are retained.</li> <li>• Entries that are present in the XML file, but not in the system, are added to the database.</li> <li>• Entries in the XML file overwrite identical entries in the database.</li> </ul>
Ignore wrong table type	<p>Activate this option to allow upload of tables with a delivery class other than C or S, or tables that have delivery class C or S but no CLIENT field. Enter these tables in the <b>Special tables</b> field. This feature should be used with care.</p>

### Compare

Setting	Description
Ignore field 'project'	<p>Activate this option to exclude the project ID from the comparison (recommended). If this option is deactivated, the compared project configurations are considered to be different even if they are identical apart from the project ID.</p>
Include system tables	<p>Activate this option to include system tables as well as customized tables in the comparison.</p> <p>If you are comparing XML files, the system tables must have been included when the configuration was downloaded to the XML file (see <a href="#">Include system tables</a>).</p>
Element 1, Element 2	<p>Select the elements that you want to compare. For example, to compare a stored XML file with a project configuration in the system:</p> <p>In the Element 1 section, select Local file and enter the file path and name.</p> <p>In the Element 2 section, select Customizing on server and select a project.</p>

When you compare configurations, the differences are displayed in the ABAP Splitscreen Editor. See the SAP ABAP documentation for information on working with this editor.

Configuration of project:		Configuration of project: HT	
4323		4323	0000000041 0000000036 000005 SO  50
4324		4324	
4325	/EBY/PDB0_CWLSCT	4325	/EBY/PDB0_CWLSCT
4326	0000000001 E	4326	0000000001 E
4327	0000000002 E  With error	4327	0000000002 E  Requisitions
4328	0000000003 E  Sales Order	4328	0000000003 E  Goods Receipts
4329	0000000004 E  Processed	4329	0000000004 E  Ordered
4330	0000000025 E  Processed requests	4330	0000000005 E  To approve
4331	0000000026 E  New requests	4331	0000000006 E  My Requisitions
4332	0000000027 E  Vendor attachments	4332	0000000007 E  Drafts
4333	0000000028 E  Accounting View Maintenance	4333	0000000008 E  Approved
4334	0000000029 E  Purchasing View Maintenance	4334	0000000009 E  In approval
4335	0000000030 E  Other	4335	0000000010 E  Not posted
4336	0000000031 E  Travel Expense, Old Prototype	4336	0000000011 E  Posted
4337	0000000032 E  Submitted	4337	0000000012 E  With errors
4338	0000000033 E  New	4338	0000000013 E  In workflow

# Configure an umbrella system

## What is an umbrella system?

An umbrella system allows you to provide a single user interface to display documents from multiple systems. Currently, the Process Director umbrella solution supports multiple SAP systems that have Process Director and Kofax Process Director Accounts Payable installed. The umbrella solution is currently only available in the SAP GUI and only for Accounts Payable documents. A separate license is required for umbrella systems (activation license only).

### Minimum requirements

Umbrella system: Kofax Process Director 7.4 and Kofax Process Director Accounts Payable 7.4

Remote systems: Kofax Process Director 7.3 Service Pack 2 or Kofax Process Director Accounts Payable 7.2 Service Pack 4

### RFC user

In the umbrella system you need an RFC user with a role containing the authorization object S\_RFC. Add the following settings to the role authorizations.

Setting	Description
Activity	16 (Execute)
Name of RFC	Add the following RFC function groups: <ul style="list-style-type: none"><li>• /EBY/PDUM_RFC</li><li>• /EBY/ICIV_UM_RFC</li><li>• /EBY/ICWC_UM_RFC</li><li>• /EBY/PDUM_WC_RFC</li></ul> The following function groups are also required in order that search helps in the remote system can be called from within the umbrella system (elementary search helps only). <ul style="list-style-type: none"><li>• /EBY/_SHLP_RFC</li><li>• /EBY/PDUM_SHLP_EXIT</li></ul>
Type of RFC object	FUGR (Function group)

### RFC destinations

In the umbrella system in `SM59` you must set up an RFC destination for each remote system that will be accessed by the umbrella system. Unless you are using trusted RFCs, it is recommended that you create two RFC destinations per system:

- A destination for "background" processing to enable users to view documents of this system in the umbrella system. Access is via the umbrella RFC user.
- A destination for "online/dialog" processing with the Current User flag set. Transactional processing such as jumping to a document or posting a document should be performed by real users, not the RFC user. Setting this flag will force users to log on with their own credentials in order to process the document. They only need to log on to a remote system once per session to process documents of that system.

## Configure an umbrella system

To configure an umbrella system, complete the following steps.

1. Make sure the prerequisites are fulfilled. For details, see [What is an umbrella system?](#)
2. In the expert IMG, click **Change system settings** > **Umbrella Solution** > **Umbrella source systems** (`/EBY/PDUM_VSYSC`).
3. In change mode, add a new entry.
4. Enter a system ID for the remote system whose documents should be available in the umbrella system and a corresponding [background RFC destination](#). We recommend using trusted RFCs so that users do not have to log on when accessing remote systems. See the SAP documentation for information on trusted RFCs.

**Tip** You can click the **Detail view**  button to view and edit the RFC destination.

5. (Optional, only if not using trusted RFCs). Enter a [dialog/online RFC destination](#). This will force users to log on to process documents. Within a session, log on is only necessary for the first processing action for that system.
6. Repeat steps 3 to 5 for each remote system that should be available in the umbrella system.
7. Click the **Save**  button.
8. Run the `/EBY/ICIV_DOC_SYNCHRONIZE` program for each remote system to make the documents available in the umbrella system.

After the initial import of existing documents, you can create and schedule variants for each remote system and use a dynamic document date to import new documents from the remote system. For example, schedule the program to run in the early hours of the morning and set the document date to the current date -1 using the variant selection variables. This will bring all documents created on the previous day into the umbrella system.

## Configure remote function modules

The default system settings for the umbrella solution specify which function modules require the user to log on in order to process a document. For example, to display the invoice or the PO document, the user must log on. Within a session, log on is only necessary for the first processing action for that system.

**Important** Do not make changes in the default system settings.

You can override the default system settings for the dialog remote function modules.

For example, if you do not want to force log on for displaying the workflow status, add the /EBY/ICIV\_UM\_RFC\_WC\_STATUS\_DIS function module and check the Remove option.

You can add any function modules from the following function groups:

- /EBY/PDUM\_RFC
- /EBY/ICIV\_UM\_RFC
- /EBY/ICWC\_UM\_RFC
- /EBY/PDUM\_WC\_RFC

To configure remote function modules, complete the following steps.

1. Go to /EBY/PDUM\_VFMDC (Expert IMG > **Change system settings** > **Umbrella Solution** > **Dialog Remote Function Modules** ).
2. In change mode, add a new entry.
3. Enter the remote system ID or leave blank if the setting should apply for all remote systems.
4. Enter the function module name.
5. To disable a default function module setting, check **Remove**
6. Click the **Save**  button.

## Map umbrella workflow descriptions

Workflows may have different descriptions in different systems, making it difficult for the processor to decide which workflow to choose. You can therefore map the workflow descriptions of the remote systems to a unified workflow description in the umbrella system.

The unified workflow description is only displayed on workflow start and not, for example, when forwarding a workflow step.

To map workflow descriptions, complete the following steps.

1. In the expert IMG, click **Change system settings** > **Umbrella Solution** > **Workflow description mapping** (/EBY/PDUM\_VWCMC).
2. Enter the system ID of the remote system and the workflow ID of the remote system workflow.
3. In the **Workflow description** field, enter the unified description.
4. Repeat steps 2 and 3 for all remote workflow IDs that should be mapped to this unified workflow description.
5. Click the **Save**  button.

## Use centralized umbrella workflows

Instead of [mapping workflow descriptions](#) between the umbrella system and the remote system, you can define centralized workflows in the umbrella system. In this case, workflows defined in the remote systems are not available.

As soon as you have activated at least one workflow for Incoming Invoices in the umbrella system, all workflows in the remote systems are no longer accessible.

Centralized umbrella workflows are defined in the `/EBY` namespace. If you wish to use custom logic to modify the workflow, you must use the Workflow handling, Workflow steps handling or Workflow email handling BAdI instead of a User Exit.

**Note** The use of centralized workflows requires a global workflow license (WC).

To use centralized workflows, complete the following steps.

1. In the umbrella system, in the **IV Incoming Invoices IMG**, click **Initial settings > Workflow**.
2. Configure and activate the workflows that you want to use.

## Map user IDs

You can map user IDs in the central umbrella system to user IDs in the remote systems. This may be useful if the same users have different user IDs in the umbrella system and the connected remote systems. If the umbrella user changes or processes a document, the name of the mapped remote system user is displayed in the message logs and the "Last modified by" field instead of the RFC user.

1. Go to transaction `/EBY/PDUM_VUSMAP`.
2. In change mode, add a new entry.
3. Enter the SAP system ID of the remote system, the user ID of the umbrella user, and the user ID of the remote user.
4. Click **Save** .

# Customize the SAP GUI detail screen

## Configure an SAP GUI static tab

You can customize the tabs in the SAP GUI document detail screen. For example, you can replace or hide the standard tabs or add your own customized tabs. Customized tabs can be statically defined or generated dynamically at runtime.

**Note** You can only create one custom runtime generated tab. You can create up to 12 statically defined tabs.

To create statically defined tabs, you can use an existing screen from the standard function group or you can define a static detail screen in a customer function group. To define a screen in a customer function group, you must also define a customer handler class.

To create a static tab, complete the following steps.

1. Create a customer function group by copying the standard function group in SE80. The standard function group will be called `/EBY/SAPLPDxx_VIEW_SCREEN`, `EBY/PDXX_DETAILS_SCREEN`, or similar and will be contained in the `/EBY/PDXX/VIEW` package (where `XX` stands for the document object type).
2. Create a screen in the newly created function group and add the necessary fields.
3. Rename the handler function module called `/EBY/PDXX_VIEW_SET_HANDLER`, `/EBY/PDXX_DETAILS_SCREEN`, or similar, in the newly created function group to a customer name.
4. Redefine the `/EBY/CL_PDVI_DETAIL_SUBSC` class or, preferably, the existing detail screen handler class (which inherits from `/EBY/CL_PDVI_DETAIL_SUBSC`) for the document object type, for example, `/EBY/CL_PDPO_DETAIL_SUBSC`.
5. Put the renamed function module from step 3 in the redefined `/EBY/IF_PDVI_DETAILS_SCREEN_HANDLER~SET_HANDLER` method of the class from step 4.
6. In the `/EBY/PDVI_VDSTC` transaction (`/EBY/PDVI_CDSTC` table), in the detail screen tab configuration, enter the customer function group and handler class.

## Configure the SAP GUI background screen

The Process Director SAP GUI background screen determines the amount of space available for the tab pages in the header data area. You can increase or reduce the space available by selecting from pre-defined screen layouts.

To configure the SAP GUI background screen, complete the following steps.

1. In the expert IMG, click **Change system settings** > **Presentation and interface** > **SAP GUI detail screen** > **Background detail screen** (`/EBY/PDVI_VDBSC`).

2. In change mode, add a new entry.
3. Enter the program name, `/EBY/SAPLPDVI_SCREEN`.
4. Use search help to select a layout. You can choose a screen layout with tabs or one without tabs.
5. Click the **Save**  button.

## Configure the SAP GUI basic screen

The Process Director SAP GUI basic screen displays fields above the header data tab pages. By default, the document status, document number and processor are displayed, but you can add other fields.

The fields you add must be available in the header data view model field catalog. If you want to add custom fields, you must first add these to the appropriate Process Director interface structure. See *Add a customer specific fields* in the *Process Director Configuration Guide* for more information.

To add a field to the basic screen, complete the following steps.

1. In transaction `SE80`, create a custom function group by copying the standard function group `/EBY/PDVI_BASICSCREEN`.
2. Copy the `/EBY/PDVI_BASICSCREEN_SET_HL` function module to a custom function module.
3. Add your field to screen 910, or create your own screen and add the field to it. Check the **Output only** option in the **General attributes** tab if the field should be read-only.
4. In the TOP include of your new function group:
  - a. Replace the standard include name in the `con_repid` constant with the name of the function group you just created.
  - b. Add the structure that contains your field to the TABLES statement next to the `/EBY/PDVI_SBASIC_BO` structure.
5. Save and activate.
6. Create a new handler class that inherits from the `/EBY/CL_PDVI_SCREEN_BASIC` class.
7. In the **Methods** tab, add a new entry **CONSTRUCTOR**, making sure that the visibility is set to **Public**.
8. Double-click the new **CONSTRUCTOR** method and add these entries:  
`mc_repid = your function group`  
`mc_var_header = your document model structure`
9. In the expert IMG, click **> Change system settings > Presentation and interface > SAP GUI detail screen > Basic screen** (`/EBY/PDVI_VDBSC`).
10. In change mode, add a new entry.
11. Enter the name of your function group, screen 0910 (or your own screen number if you created a new one) and your new handler class.
12. Click the **Save**  button.

# Customize menus and buttons

## Customize Web Application buttons

You can customize the actions (that is, the buttons) that are available in the Process Director Web Application. For example, you can add or remove buttons, and change the label, icon, tooltip and position of the buttons on the **Actions** bar.

**Note** These settings determine the general availability of an action button. You can also exclude actions for a specific document status, for workflows in general, or for a specific workflow step.

**Note** You can also create custom buttons for the Web Application that connect to an external web server instead of a Process Director action. See the *Process Director Web Application Configuration Guide* for details.

To customize the actions, complete the following steps.

1. Go to [/EBY/PDWA\\_CACTC](#) (Expert IMG > **Change system settings** > **Web Application** > **Available actions** ).
2. In change mode, add a new entry.
3. Enter the settings for the action.
4. Click the **Save**  button.

## Available actions for Web Application settings

[/EBY/PDWA\\_CACTC](#)

[Expert IMG](#) > [Change system settings](#) > [Web Application](#) > [Available actions](#)

Setting	Description
Object	Process type to which the setting applies. Left blank, the setting applies to all process types.
Environment	The environment in which the settings apply.
Component	Area of the GUI to which the setting applies.
Edit mode	Editing mode to which the setting applies.
Action	Action to which the setting applies.
Description	Text displayed on the button.
Tooltip	Tooltip displayed when the mouse hovers over the button.
State	The document status to which the settings apply.

Setting	Description
Icon	Icon displayed on the button.
Icon name	SAP icon name (automatically entered).
Remove	Select to deactivate an existing Default system settings entry for the action (that is, to remove the action button).
Order	Position in which the button will appear on the Actions bar.

## Examples

### Add an action button

This example adds the **Reject document** button to the Web Application Actions bar for Customer Orders in the document overview list.

Available actions for Web application												
Object	Env.	Component	Edit mode	Action	Description	Tooltip	S..	Icon	Icon	Icon name	R..	Order
SO Customer Order		OVERVIEW...	Display mode	REJEC2 Reject document	Reject doc.	Reject document2		@30@			<input type="checkbox"/>	200

Customer Orders (1627)											

### Remove an action button

This example removes the **Post** button from the Web Application Actions bar for Requisitions in the document detail view.

**Note** You must check the **Remove** option to deactivate the default system setting for this button.

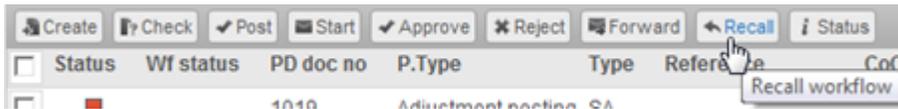
Available actions for Web application												
Object	Env.	Component	Edit mode	Action	Desc...	Tooltip	S..	Icon	Icon	Icon na...	Remove	Order
PO Requisition		DETAIL ..	X Change mode	POST_CHK_CHANGE			2				<input checked="" type="checkbox"/>	130

### Change label, tooltip and icon

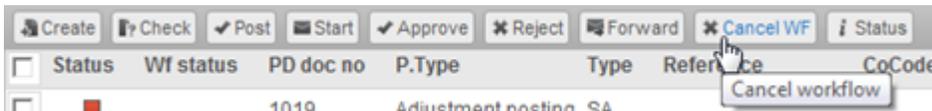
This example changes the button label, icon and tooltip text for the workflow recall button in the document overview list and the document detail view. Since the **Object** field is blank, it applies for all process types.

Available actions for Web application												
Object	Env.	Component	Edit mode	Action	Description	Tooltip	S..	Icon	Icon	Ico...	Repl./Rem.	Order
		OVERVIE..	Display mode	WCRECALLOINITCHECK	Bulk workflow recall	Cancel WF	3	@02@			<input type="checkbox"/>	80
		DETAIL ..	Display mode	WCRECALLOCHECK_DOC	Check doc. before recall (read)	Cancel WF	0	@02@			<input type="checkbox"/>	130
		DETAIL ..	X Change mode	WCRECALLS1CHECK_DOC	Check doc. before recall (edi...	Cancel WF	0	@02@			<input type="checkbox"/>	130

Before



After



## Change position

This example positions the **Status** button between the **Start** and **Approve** buttons on the Actions bar in the document overview list. Since the **Object** field is blank, it applies for all process types.

To change the position of the **Status** button, complete the following steps.

1. Check in **Default system settings > Web Application > Available actions** the standard position (in the **Order** field) of the buttons **Start** and **Approve**. Do not make changes here!

Object	Env.	Component	Edit mode	Action	Description	S..	Icon	Icon	Icon name	O...
		OVERVIE...	Display mode	WCAPPROVE0INITCHECK Bulk workflow approve	Approve	3	✓	✓	ICON_CHECKED	50
		OVERVIE...	Display mode	WCFORWARD0INITCHECK Bulk workflow forward	Forward	3	↶	↷	ICON_SYSTEM_REDO	70
		OVERVIE...	Display mode	WCRECALLOINITCHECK Bulk workflow recall	Recall	3	↶	↷	ICON_SYSTEM_UNDO	80
		OVERVIE...	Display mode	WCREJECT0INITCHECK Bulk workflow reject	Reject	3	✗	✗	ICON_INCOMPLETE	60
		OVERVIE...	Display mode	WCSTART0INITCHECK Bulk workflow start	Start	3	▶	▶	ICON_MAIL	40
		OVERVIE...	Display mode	WCSTATUSSHOW_LIST Show workflow status -...	Status	2	i	i	ICON_INFORMATION	180

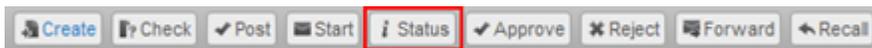
2. For the **Status** button, assign an **Order** value between 40 and 50 in **Change system settings > Web Application > Available actions**.

Object	Env.	Component	Edit mode	Action	Description	Tooltip	S..	Icon	Icon	Ico...	Repl./Rem.	Order
		OVERVIEW...	Display mode	WCSTATUSSHOW_LIST Show workflow status - List	Status	Workflow status 2	@0S@				<input type="checkbox"/>	45

Before



After



## Customize SAP GUI buttons

You can customize the buttons that are available on the Process Director SAP GUI Application toolbar. For example, you can add or remove buttons and change the label, icon, tooltip and position of the buttons.

Process Director provides a standard toolbar configuration for each standard process type. Not all available buttons are displayed for all process types by default. For example, the standard buttons Create and Copy are not displayed by default for Customer Orders. If you want users to be able to create and copy Customer Orders, you must add these buttons to the custom toolbar. You can also add custom buttons, that is, buttons for functions that are not available by default.

First, you will need to find the appropriate function codes for the button that you want to change in the standard menu structure:

1. Go to /EBY/PDVI\_VMEN (Expert IMG **Default system settings** > **Presentation and interface** > **Menu structure, user commands and object type assignment** > **Menu structure and user commands** ).
2. Find the toolbar function code for the process type that you want to change. This is usually XX\_BUT\_CHANGE, whereby XX is the ID of the process type, for example, PO\_BUT\_CHANGE for requisitions, or DN\_BUT\_CHANGE for Goods Receipts.
3. Find the button function code. You can easily recognize button function codes because they have no reference function code.
4. Make a note of these function codes.

PD menu structure, entries and functions								
Function code	Ref. function code	M.	Type	F key number	F..	Text	Icon	Icon
FILEUPLIST		66	Normal appl...	77	Ctrl-Inse...	File upload fro...		
FINALIZE		0	Normal appl...					
FI_BUT_CHANGE	CANCEL	0	Normal appl...					
FI_BUT_CHANGE	CHANGE_POST_TY...	0	Normal appl...					
FI_BUT_CHANGE	CHECK	0	Normal appl...					

You can then make changes:

5. Go to /EBY/PDVI\_VMENC (Expert IMG > **Change system settings** > **Presentation and interface** > **Custom menu structure, user commands and object type assignment** > **Menu structure and user commands** ).
6. In change mode, add a new entry.
7. Enter the settings.
8. Click the **Save** button.

## Examples

### Add a standard button

This example adds the Create button to the Customer Orders toolbar.

1. In the **Function code** field, enter the function code for the toolbar, in this example, SO\_BUT\_CHANGE.
2. In the **Ref. func. code** field, enter the function code for the button, in this example, CREATE.  
Note that these are the only fields you need to fill; all other settings will be inherited from the standard system entry.
3. Click the **Save** button.

4. Go to PDVI\_VUCOC ( **Change system settings** > **Presentation and interface** > **Assign function codes to actions** ).
5. In change mode, add a new entry.
6. Select the appropriate component and display mode.
7. Enter the function code that you noted down in step 1.
8. Select the appropriate action.
9. Click the **Save**  button.

## Hide a button

1. In change mode, add a new entry.
2. In the **Function code** field, enter the function code for the toolbar, for example, PO\_BUT\_CHANGE.
3. In the **Ref. func. code** field, enter the function code that you noted down in step 1.
4. Check **Replace/Remove**.
5. Click the **Save**  button.

# Develop BAdIs

## What is a BAdI?

Process Director provides a number of [predefined BAdIs](#) as extension points for customer coding. This chapter explains how to create a BAdI implementation and describes the platform BAdIs, that is, those that apply to all process types. For information on process-specific BAdIs, see the *Process Director Reference Guide*.

BAdIs are a new SAP enhancement technology based on ABAP Objects. Like User Exits, they enable customers to add their own functionality without modifying the standard program. BAdIs have two views: the definition view and the implementation view. In the implementation view, users of BAdIs can create their own implementations that will be executed via enhancement calls from within the standard.

BAdI methods are just as easy to use as the function module based User Exits you may already be familiar with from Kofax Process Director Accounts Payable, but offer more power and flexibility, as they are based on the more modern Business Add-In (BAdI) technology. For example, they allow for multi-level system landscapes (SAP, partner, customer, country, industry, etc. versions) and may be reusable, that is, they can be used actively by any number of customers at the same time. They can also be defined according to filter values.

The actual program code is enhanced using ABAP Objects, but you do not need to understand ABAP Objects in order to use BAdIs. You can use standard procedural ABAP programming, which you add to the appropriate BAdI method. Note, though, that ABAP Objects class methods have stricter syntax rules than ABAP code outside of classes, so you should take care not to use any obsolete language constructs in your implementation coding. We recommend that you read the [Best practice guidelines for BAdIs](#) before you create your first BAdI implementation.

## User Exits/BAdIs

Process Director provides the following predefined user exits / BAdIs as extension points for customer coding. In addition to these standard platform user exits / BAdIs, process type specific user exits / BAdIs are available for individual process types.

See the *Process Director Reference Guide* for detailed information on the available BAdIs.

### Basic

BAdI	Description
Screen PBO/PAI	Add custom logic for the screen display. For example, you can use this BAdI to display or verify values.

BAdI	Description
Modify Worklist	Add custom logic for the Process Director Worklist configuration.
Modify field profiles	Add custom logic when field statuses are applied.
Set Worklist node of a single document link	Add custom logic to open a specific Worklist node when a user clicks a single document link in a workflow notification email or SAP Business Workflow work item.
Determination results	Add custom logic after determinations have been performed.
Rejection reason	Add custom logic before and after a rejection reason is entered.

### Web Application

BAdI	Description
Set messages	Add custom logic to modify messages sent to the Web Application.
Set actions	Add custom logic to modify which <a href="#">actions are available</a> to the Web Application.
User logon	Add custom logic to modify the behavior of the application during logon to the Web Application.

### Workflow

BAdI	Description
Workflow email handling	Add custom logic for handling workflow emails.
Workflow handling	Add custom logic for handling workflows.
Workflow steps handling	Add custom logic for handling workflow steps.

## Create a BAdI implementation

To create a user exit/BAdI implementation, complete the following steps.

1. In the process type IMG, expand the activity **User exits / BAdIs**.
2. Select the user exit / BAdI that you want to implement.
3. Click the **Create Implementation**  button.
4. In the **Information** popup, click the **Continue**  button.
5. In the **Create Enhancement Implementation** dialog box, enter a name for the enhancement implementation and a short description for your implementation of the BAdI and click the **Continue**  button.

**Note** An enhancement implementation is a container for BAdI implementations.

6. In the **Create Object Directory Entry** dialog box, enter the name of the package and click the **Save (Enter)**  button.
7. In the **Create BAdI Implementations for Existing BAdI Definitions** table, enter the name of the BAdI implementation and its implementation class.
8. Click the **Continue**  button.
9. In the **Enhancement Implementation Change** dialog box, for your BAdI implementation, select the **Implementing Class** option to display the list of methods.
10. In the right pane, double-click the method you want to use.

Implementing Class	
Interface	/EBY/IF_PDVI_BADI_SCREEN
Implementing Class	ZEBY_CL_PDVI_SCREEN_UT_IMP  
Method	Short description
<input type="checkbox"/> /EBY/IF_PDVI_BADI_SCREEN~PBO	Process Before Output
<input checked="" type="checkbox"/> /EBY/IF_PDVI_BADI_SCREEN~PAI	Process After Input
<input type="checkbox"/> /EBY/IF_PDVI_BADI_SCREEN~SET_PAI_OKCODE	Set OKCODE in Process After Input
<input type="checkbox"/> /EBY/IF_PDVI_BADI_SCREEN~EXCLUDE_FUNCTION_CODES	Modify excluded function codes

11. Add your coding.

Method	/EBY/IF_PDVI_BADI_SCREEN~PAI	Active
1	<code>METHOD /eby/if_pdvi_badi_screen-pai.</code>	
2		
3	<code>    CHECK mb_test = abap_true.</code>	
4		
5	<code>    mb_called = abap_true.</code>	
6	<code>    cc_okcode = 'TEST_OKCODE'.</code>	
7		
8	<code>ENDMETHOD.</code>	
9		

12. Save your changes and click the **Activate**  button to activate the BAdI.

## Best practice guidelines for BAdIs

### General best practice guidelines

**Important** Specific user exits / BAdIs may have additional guidelines of their own. However, these general rules apply to the coding for all BAdI implementations:

Specific user exits / BAdIs may have additional guidelines of their own. However, these general rules apply to the coding for all BAdI implementations:

### Use filter-dependent BAdIs

For filter-dependent BAdIs, filters can be defined on literal field values and/or [CP patterns](#).

To preserve compatibility with SAP 4.6C systems, Process Director does not support wildcards (\*) in filter fields. This means that if you need a BAdI that supports all Process Director object types, you must create a BAdI implementation for each object type.

### Issue messages

- To issue messages, your coding should use the static methods provided for this purpose by the class /EBY/CL\_PDBO\_EXC, such as ADD\_MSG or ADD\_BAPIMSG. These are similar to the helper function module /COCKPIT/OBJ\_MESSAGE\_APPEND in INVOICE COCKPIT. Use the constants provided as /EBY/CL\_PDBO\_EXC class attributes to set the appropriate message group.
- After calling a static wrapper method to issue a message, always add a dummy entry for the message. This ensures that the message is correctly cross-referenced.
- Keep in mind that although you can issue messages, you cannot read messages that have been previously issued. That is, Process Director user exits / BAdIs do not offer a functional equivalent to the CT\_MESSAGES parameter in INVOICE\_COCKPIT user exits.

### Make BAPI calls

Whenever possible, your coding should use the convenience methods provided by the wrapper class /EBY/CL\_BAPI to make BAPI calls.

### Access customer-specific fields

Do not directly access customer-specific fields. Instead, use the MOVE-CORRESPONDING pattern.

### Write a document back to the database

To write a document back to the database in your user exit / BAdI implementation coding, you can call /EBY/CL\_PDBO\_SESSION=>MR\_SESSION->UPDATE, passing the document's IR\_OBJECT reference. This reference is usually exposed to the user exit /BAdI method as an IMPORTING parameter.

**Important note:** Any data modifications you have made to CHANGING parameters will *not* be written back to the database at this point. This is due to the fact that document object model structures passed in as CHANGING parameters are copied back to the object by the Process Director user exit / BAdI framework after the user exit / BAdI method exits. Therefore, if you want your data modifications to be written back to the database, you must make them via the IR\_OBJECT reference.

### Access the Process Director runtime context

- The user session context is available everywhere via the static attribute MR\_CONTEXT of the class /EBY/CL\_PDBO\_CONTEXT.
- If you use standalone helper classes in your coding, put them in an independent package. If you do not want to instantiate a helper object every time the user exit / BAdI method is executed, you can save an object reference to the user session context using the /EBY/CL\_PDBO\_CONTEXT instance method SET\_ATTR\_DEF.
- To run a Process Director action, call the /EBY/CL\_PDBO\_CONTROLLER instance method RUN1, passing the action logical name as the importing parameter IC\_ACTION. For example, you can create a new document by passing the action logical name CREAT1.  
Test and support
- You may not do customer development in the /EBY/ namespace.
- Keep a copy of all the customer-specific developments. This will help you to successfully escalate support cases to third-level support.

- Do not rely on multiple use BAdI implementations being called in a specific order.
- If you have customer logic that is amount related, always test it with JPY and any other currencies that might be used by the customer.
- Equivalent of Kofax Process Director Accounts Payable shadow customizing in Process Director? Use a CHECK sy-uname = '...' statement at the beginning of each method body.

# Create custom rules

## About custom rules

Process Director provides the following class templates to assist you in creating your own custom presets, checks and determinations.

Template name	Description
/EBY/CL_PDBO_PRESET_TEMPL	Template for creating custom presets
/EBY/CL_PDBO_EVT_CHK_TEMPL	Template for creating custom checks
/EBY/CL_PDBO_DET_INIT_TEMPL	Template for creating custom determinations
/EBY/CL_PDBO_DET_TEMPL	Template for creating custom determination searches

These templates provide pre-defined methods to which you can add your own coding. Guidance is provided in the template coding in the form of commented explanations, instructions and examples.

Ty.	Parameter	Type spec.	Description
>	IR_OBJECT	TYPE REF TO /EBY/IF_OBJECT	The object to check
>	IS_EVENT	TYPE /EBY/PDBO_SPRO	Configuration of processes (possibly to get evt parameters)
!	ERROR_OCCURRED		

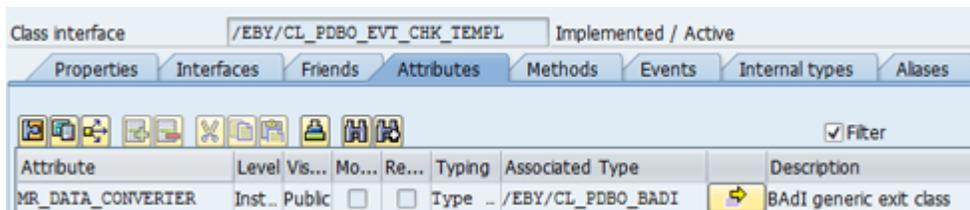
  

```

Method: CHECK Activ
1 METHOD check .
2
3
4 *-----*
5 * If your document type has a data converter available you can access
6 * all of its data using structures/tables from the data converter.
7 *
8 * In our example that would be:
9 *
10 * lr_data_converter->ms_po_hdr
11 * lr_data_converter->mt_items
12 * lr_data_converter->mt_item_accounts
13 * lr_data_converter->mt_item_schedules
14 * lr_data_converter->mt_item_conditions
15 * lr_data_converter->mt_partners
16 *-----*
17
18 * Example of check
19 * Very important !!!
20 * If a check should fail only create an error message
21 * Always use /eby/cl_pdbo_exc=>con_msggrp_check as the PD message group
22 * Don't create a warning. An error can always be converted to a warning
23 * Don't raise the exception ERROR_OCCURRED. This should be done only
24 * when something unexpected happened
25
26 * >>> Put your code here
27 * DATA:
28 *   ls_item LIKE LINE OF mr_data_converter->mt_items.
29 *
30 * LOOP AT mr_data_converter->mt_items INTO ls_item.
31 *   IF mr_data_converter->ms_po_hdr-currency = 'EUR'
32 *     AND ls_item-quantity = 10.
33 *     CALL METHOD /eby/cl_pdbo_exc=>add_msg
  
```

### Data converters

Along with the templates, Process Director provides data converters to simplify document data handling. A data converter is available for each standard process type. It is named /EBY/CL\_PDx\_BADI, whereby x is the ID of the process type, for example, /EBY/CL\_PDPO\_BADI for the Requisitions data converter. If a data converter is not available for a process type (for example, for custom process types), you can use the generic data converter, /EBY/CL\_PDBO\_BADI. You specify which data converter you want to use in the **Attributes** tab of the class:



### Create a preset ID

Should the system IDs not be sufficient for your purposes, you can also define customer preset IDs:

1. In change mode, add a new entry.
2. Enter a preset ID and preset description.
3. Save your settings.
4. Go to transaction `/EBY/PDBO_VPSVC` and assign preset values to the new preset ID.

## Create a custom preset

Process Director provides the `/EBY/CL_PDBO_PRESET_TEMPL` preset class template to assist you in developing your own presets.

The presets configuration program requires that a preset class is tagged with an interface. Process Director provides a tagging interface for all standard process types. It is named `/EBY/IF_PD $x$ _PRESET`, whereby  $x$  is the ID of the process type, for example, `/EBY/IF_PDPO_PRESET` for the Requisitions tagging interface. If a tagging interface is not available for a process type (for example, for custom process types), you must create this interface and add it to the Interfaces tab of the class.

To create your own preset class, complete the following steps.

1. In `SE24`, copy the `/EBY/CL_PDBO_PRESET_TEMPL` class and add your own description.
2. In the **Interfaces** tab of your new class, add the appropriate tagging interface.
3. In the **Attributes** tab of your new class, enter the appropriate [data converter](#) in the **Associated Type** field of the `MR_DATA_CONVERTER` attribute.

4. In the **Methods** tab, double-click the PROCESS method and make your coding changes. Make sure that you uncomment the appropriate section, depending on which data converter you are using.

Method	/EBY/IF_PDBO_PRESET-PROCESS	Activ
29	*-----*	
30	* If your document type has a data converter available leave that part	
31	* of the code uncommented:	
32		
33	<input type="checkbox"/> * CALL METHOD specific_data_converter	
34	* EXPORTING	
35	* ir_object = ir_object	
36	* ir_context = ir_context	
37	* is_preset = is_preset	
38	* CHANGING	
39	* c_value = c_value	
40	* cs_values = cs_values	
41	* EXCEPTIONS	
42	* OTHERS = 1.	
43	* IF sy-subrc <> 0. RAISE error_occurred. ENDIF.	
44		
45	<input type="checkbox"/> *-----*	
46	* If your document type does not have a data converter available	
47	* uncomment the code below and comment the code above:	
48		
49	<input type="checkbox"/> * CALL METHOD generic_data_converter	
50	* EXPORTING	
51	* ir_object = ir_object	
52	* ir_context = ir_context	
53	* is_preset = is_preset	
54	* CHANGING	
55	* c_value = c_value	
56	* cs_values = cs_values	
57	* EXCEPTIONS	
58	* OTHERS = 1.	
59	* IF sy-subrc <> 0. RAISE error_occurred. ENDIF.	
60		
61	*-----*	

5. Save and activate the class.

After you have created your preset, you can add it to the presets configuration. See the *Process Director Configuration Guide* for more information.

## Create a custom check

Process Director provides the /EBY/CL\_PDBO\_EVT\_CHK\_TEMPL check class template to assist you in developing your own checks.

The checks configuration program requires that a check class is tagged with an interface. Process Director provides a tagging interface for all standard process types. It is named /EBY/IF\_PD<sub>x</sub>\_CHK, where x is the ID of the process type, for example, /EBY/IF\_PDPO\_CHK for the Requisitions tagging interface. If

a tagging interface is not available for a process type (for example, for custom process types), you must create this interface and add it to the Interfaces tab of the class.

To create your own check class, complete the following steps.

1. In SE24, copy the `/EBY/CL_PDBO_EVT_CHK_TEMPL` class and add your own description.
2. In the **Interfaces** tab of your new class, add the appropriate tagging interface.
3. In the **Attributes** tab of your new class, enter the appropriate [data converter](#) in the **Associated Type** field of the `MR_DATA_CONVERTER` attribute.
4. In the **Methods** tab, double-click the method `CHECK` and redefine it in accordance with customer requirements.
5. If you want to perform the check only for specific documents or documents sublevels, redefine the `GET_OBJECTS` method accordingly. In most cases, this is not necessary.
6. Save and activate the class.

After you have created your check, you can add it to the checks configuration. See the *Process Director Configuration Guide* for more information.

**Important** Always use `/eby/cl_pdbo_exc=>con_msggrp_check` as the PD message group. If a check should fail, only create an error message. Do not create a warning. Errors can always be converted to a warning in the check configuration settings. Do not raise the exception `ERROR_OCCURRED`. This should be done only when something unexpected happens.

## Create a custom determination

Process Director provides the `/EBY/CL_PDBO_DET_INIT_TEMPL` and `/EBY/CL_PDBO_DET_TEMPL` determination class templates to assist you in developing your own determinations.

The determinations configuration program requires that a determination class is tagged with an interface. Process Director provides a tagging interface for all standard process types. It is named `/EBY/IF_PDx_DET`, whereby `x` is the ID of the process type, for example, `/EBY/IF_PDPO_DET` for the Requisitions tagging interface. If a tagging interface is not available for a process type (for example, for custom process types), you must create this interface and add it to the Interfaces tab of the class.

To create your own determination class, complete the following steps.

### Create the new determination

To create a new determination, complete the following steps.

1. In SE24, copy the `/EBY/CL_PDBO_DET_INIT_TEMPL` class and add your own description.
2. In the **Interfaces** tab of your new class, add the appropriate tagging interface.
3. In the **Attributes** tab of your new class, enter the appropriate [data converter](#) in the **Associated Type** field of the `MR_DATA_CONVERTER` attribute.
4. In the **Methods** tab, double-click the `GET_RESULT_TYPE` method and redefine it in accordance with customer requirements.
5. If you want to perform the determination only for specific documents or documents sublevels, redefine the `GET_OBJECTS` method accordingly. In most cases, this is not necessary.

6. Save and activate the class.

## Create a new determination search

To create a new determination search, complete the following steps.

1. In SE24, copy the `/EBY/CL_PDBO_DET_TEMPL` class and add your own description.
2. In the **Interfaces** tab of your new class, add the appropriate tagging interface.
3. In the **Attributes** tab of your new class, enter the appropriate [data converter](#) in the **Associated Type** field of the `MR_DATA_CONVERTER` attribute.
4. In the **Methods** tab, redefine the `DETERMINE` and `UPDATE` methods in accordance with customer requirements.
5. You may also want to add a message to the `CREATE_MESSAGE` method that will be displayed if a determination search returns no result or more than one result.
6. Save and activate the class.

Repeat these steps if you want to add more searches for the determination.

**Note** Ensure that the methods in the classes return the expected values and that the custom determinations are correctly listed in the maintenance transaction. If not, you need to adjust the `GET_DATA_TYPE` method by redefining it in the custom class.

After you have created your determination, you can add it to the determinations configuration. See the *Process Director Configuration Guide* for more information.

# Define a document status

Process Director provides standard document statuses for each process type. Some process types also ship with standard document substatuses. See the *Process Director Reference Guide* for detailed information on document statuses/substatuses for a specific process type.

You can add to these statuses or replace/remove them in the customer-level customizing, and define your own status icons.

## Define document status/substatus

To define a document status or substatus, complete the following steps.

1. To create a customer-specific:
  - Status, go to /EBY/PDBO\_VSTAC (expert IMG > **Change system settings** > **Model** > **Document statuses/substatuses** > **Customer document statuses** ).
  - Substatus go to /EBY/PDBO\_VSSTC (expert IMG > **Change system settings** > **Model** > **Document statuses/substatuses** > **Customer document substatuses** ).
2. In change mode, add a new entry.
3. Enter a two-character ID. To override a system status/substatus, use search help to select it, and then select the **Remove** check box.
4. Click the **Save**  button.

## Define icons for document status/substatus

To define icons for a document status or substatus, complete the following steps.

1. To define customer-specific icons for a:
  - Status, go to /EBY/PDVI\_VSTAC (expert IMG > **Change system settings** > **Presentation and interface** > **Status/substatus - define document icons** > **Document status icons** ).
  - Substatus, go to /EBY/PDVI\_VSSTC (expert IMG > **Change system settings** > **Presentation and interface** > **Status/substatus - define document icons** > **Document substatus icons** ).
2. In change mode, add a new entry.
3. Select a status/substatus and an icon.
4. Click the **Save**  button.

## Map message number to substatus code

To map a message number to a substatus code, complete the following steps.

1. Go to /EBY/PDBO\_VM2SC (expert IMG > **Change system settings** > **Model** > **Document statuses/substatuses** > **Customer Mapping message number to substatus code** ).
2. In change mode, add a new entry.
3. Fill in the required fields.

**Note** To override a system message for a substatus, use search help to select it, and then select the **Remove** check box

4. Click the **Save**  button.

# Configure a popup

You can customize some of the pop-up dialog boxes in Process Director. For example, you can configure the pop-up dialog box that opens when a user creates a new master data maintenance request.

To find out which pop-up dialog boxes you can configure for a specific process type, complete the following step.

- In the expert IMG for that process type, go to `/EBY/PDBO_VC_POPUP` ( **Default system settings** > **Other** > **Popup title, fields and dropdowns** ).

**Important** Do not make changes in the default system settings, this is for reference only.

You can change the popup title, the fields displayed in the popup and their attributes, and the values available in dropdown lists in the popup. Go to the appropriate transaction and enter the required settings.

## Change the popup title

To change the popup title, complete the following steps.

1. Go to `/EBY/PDBO_VPUIC` (expert IMG > **Change system settings** > **Other** > **Popup title, fields and dropdowns** > **Popup IDs** ).
2. Select the popup ID and enter the new popup title.

## Configure fields

To add a new field, hide a standard field, change the order of fields or the field attributes, complete the following steps.

1. Go to `/EBY/PDBO_VPUFC` (expert IMG > **Change system settings** > **Other** > **Popup title, fields and dropdowns** > **Popup fields** ).
2. Select the popup ID, enter the table name and field name.

The table name is the name of the interface structure that contains the field. Refer to the appropriate [process type document model customization](#) section in the *Process Director Reference Guide* for the correct name of the structure. For example, the interface structure for vendor master header data is `/EBY/PDMDVM_SHDR_IF`. The field must also be defined in the field catalog.

### 3. Configure the fields, as required.

- a. To add a new field, enter the appropriate settings for the field.

To add custom fields (Z-fields), add the field to the appropriate customer include and the field catalog. See the *Process Director Configuration Guide* for more information.

- b. To hide a standard field, select the **Remove** check box.

- c. To change the order of fields, in the **Order** field, enter a number.

This number is relative to the order number in the default system settings. In the example shown below, the first five fields are numbered 10-50 in the default settings. So if you want to move the BUKRS field to appear between the ACTIVITY and LIFNR fields, you must assign a number between 10 and 20.

Popup fields		
Table Name	Field Name	Order
/EBY/PDMDVM_SHDR_IF	ACTIVITY	10
/EBY/PDMDVM_SHDR_IF	BUKRS	30
/EBY/PDMDVM_SHDR_IF	EKORG	40
/EBY/PDMDVM_SHDR_IF	KTOKK	50
/EBY/PDMDVM_SHDR_IF	LIFNR	20

- d. To change field attributes, enter the appropriate settings.

## Add values to dropdown fields

To add values to dropdown fields, complete the following steps.

1. Go to /EBY/PDBO\_VPUFC (expert IMG > **Change system settings** > **Other** > **Popup title, fields and dropdowns** > **Popup fields** ).
2. Make sure the field is set to dropdown value **D Input field with drop down box**.
3. Go to /EBY/PDBO\_VPUDC (expert IMG > **Change system settings** > **Other** > **Popup title, fields and dropdowns** > **Popup dropdowns** ).
4. Add a new line for each value and specify the value and short text.

# Activate subset posting

Some document types in Process Director may contain different sets of data that are processed by different users. For example, a vendor master data maintenance request contains general, address, bank, accounting and purchasing data, and different users within the organization may be responsible for maintaining these data types in the vendor master record.

You can configure subset based posting for these document types, so that instead of posting all data at the same time, the different data types can be posted independently of one another.

To configure subset-based posting, complete the following steps.

1. In the expert IMG, click **Change system settings > Posting > Subset based posting activation** (/EBY/PDBO\_VPSSAC).
2. In change mode, add a new entry.
3. Select the **Status**, **Substatus** and **Posting** subset.
4. Click the **Save**  button.

# Create a custom process type

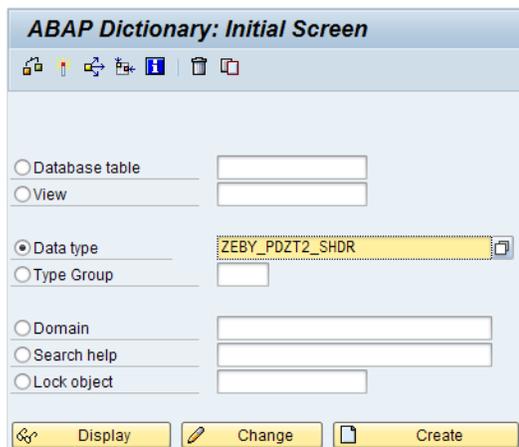
If the standard Process Director process types do not fulfill customer needs, you can create custom process types by copying the XY Generic Object Template or an existing process type.

- To create a new custom process type, go to [/EBY/PDXY\\_COPY](#).
- To modify an existing custom process type, go to [/EBY/PDBOC](#), select the custom process and click , then select the activity Create new object type or add model level.

## Create a new object type

To create a new object type, complete the following steps.

1. If you have not already done so, enter SAP transaction `SE11` and create a new Z model data structure for the object type's header level:



**ABAP Dictionary: Initial Screen**

Database table

View

Data type

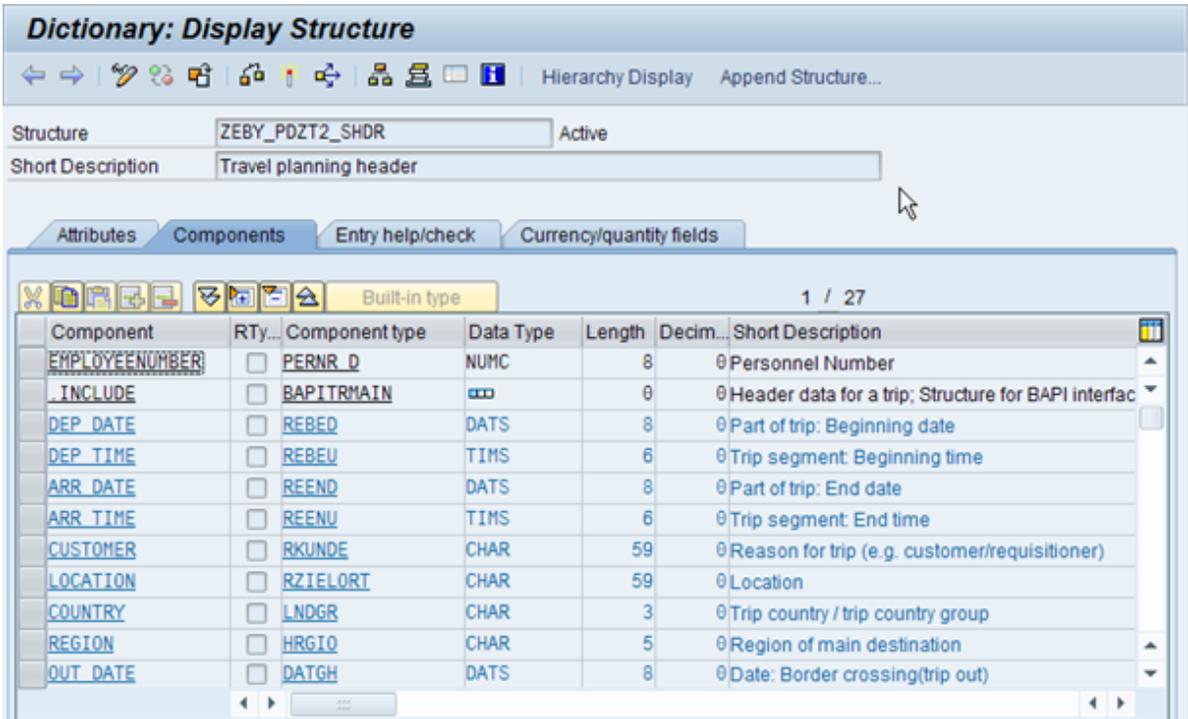
Type Group

Domain

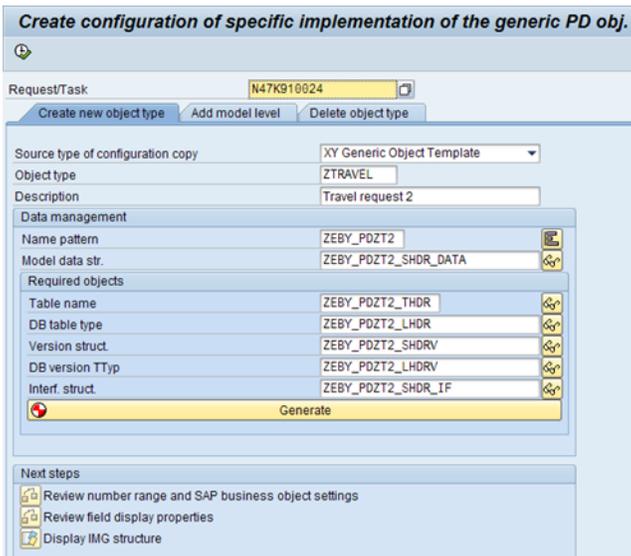
Search help

Lock object

2. Add the required components for the header level:



3. In the /EBY/PDXY\_COPY transaction, select the **Create new object type** tab.



4. Configure the [settings](#). The name pattern assists you in entering names for the required objects. Enter a pattern (this will usually be the first few letters of the name of your new model data structure) and click the **Align**  button. Names based on this pattern are entered in all fields in the **Required objects** section. You can change these names if necessary. Make sure that the model data structure name is correct after alignment.

5. Click **Generate** to generate the required DDIC objects.

**Note** You are only prompted once for a development class and a transport request, when generating the first object. The four remaining objects are also created in this development class and transport request.

6. After generating the data dictionary objects, display each one with the  button and click the  button to activate them. This example shows what a generated table looks like:

**Dictionary: Display Table**

Transp. table: ZEBY\_PDZT2\_THDR New  
 Short Description: PROCESS DIRECTOR generated structure

Attributes | Delivery and Maintenance | **Fields** | Entry help/check | Currency/Quantity Fields

Srch help Built-in type 1 / 30

Field	Key	Initi...	Data element	Data Ty...	Length	Decim...	Short Description
CLIENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MANDT	CLNT	3	0	Client
_INCLUDE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	/EBY/PDBO_SGUID	STRU	0	0	Key field of any object
..INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	ZEBY_PDZT2_SHDR	STRU	0	0	Travel planning header
EMPLOYEENUMBER	<input type="checkbox"/>	<input type="checkbox"/>	PERNR_D	NUMC	8	0	Personnel Number
_INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	BAPITRMAIN	STRU	0	0	Header data for a trip; Structure for BAPI interfa
DEP_DATE	<input type="checkbox"/>	<input type="checkbox"/>	REBED	DATS	8	0	Part of trip: Beginning date
DEP_TIME	<input type="checkbox"/>	<input type="checkbox"/>	REBEU	TIMS	6	0	Trip segment: Beginning time
ARR_DATE	<input type="checkbox"/>	<input type="checkbox"/>	REEND	DATS	8	0	Part of trip: End date
ARR_TIME	<input type="checkbox"/>	<input type="checkbox"/>	REENU	TIMS	6	0	Trip segment: End time
CUSTOMER	<input type="checkbox"/>	<input type="checkbox"/>	RKUNDE	CHAR	59	0	Reason for trip (e.g. customer/requisitioner)
LOCATION	<input type="checkbox"/>	<input type="checkbox"/>	RZIELORT	CHAR	59	0	Location

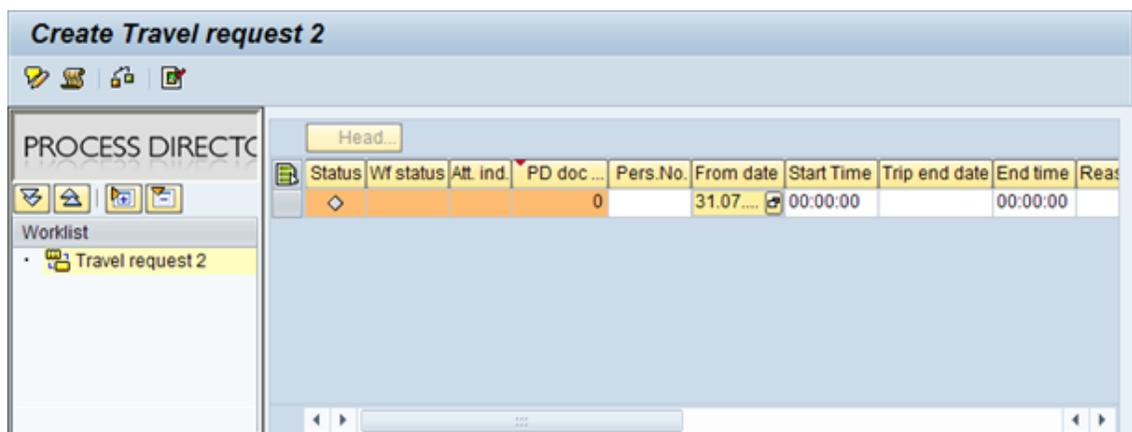
**Important** Make sure to activate all generated objects with the  button, as they are generated with the status, **New**.

7. Click the **Execute**  button to create the object type configuration.
8. **Add model levels.** (You do not need to do this for object types that only have a header level). You can create any number of levels and sublevels.

## 9. Perform the **Next steps**:

- a. Review number range and SAP business object settings. Usually the only setting you will need to change is the IMG tree text. This is the text that is displayed at the top of the IMG tree of the custom process type.
- b. Review field display properties. This takes you to the header level system field catalog. By default, all fields are set to editable — you probably need to adjust these and other settings per field.

After configuring a Worklist node for your new process type and performing other customizing steps as applicable, the process type will be available for use in `/EBY/PD`. For example, you can create new documents using the list overview:



## Add a model level

To add a model level, complete the following steps.

1. If you have not already done so, enter SAP transaction `SE11` and create a new Z model data structure for the model level.
2. In SAP transaction `/EBY/PDXY_COPY`, select the tab **Add model level**.
3. Configure the [settings](#).
4. Click **Generate** to generate the required DDIC objects.

**Note** You are only prompted once for a development class and a transport request, when generating the first object. The four remaining objects are also created in this development class and transport request.

5. After generating the data dictionary objects, display each one with the button  and click the button  to activate them.

**Important** Make sure to activate all generated objects with the button , as they are generated with status New.

## 6. Configure the visualization settings:

Optional settings	
Grid title	Enter the title that appears as the ALV grid title in SAP GUI and/or as the table label in the Web Application.
<b>Create SAP GUI Grid</b>	Check this if the model level should appear in SAP GUI. Leave this unchecked if the model level is for internal use only.
<b>Create SAP GUI Grid &gt; in a popup</b>	Check this if the model level should be displayed in a popup rather than on a grid in the main screen.
<b>Create web application grid</b>	Check this if the model level should appear in the Web Application. Leave this unchecked if the model level is for internal use only.

7. Click the **Execute**  button to create the object type configuration.

## Delete an object type

To delete an object type, complete the following steps.

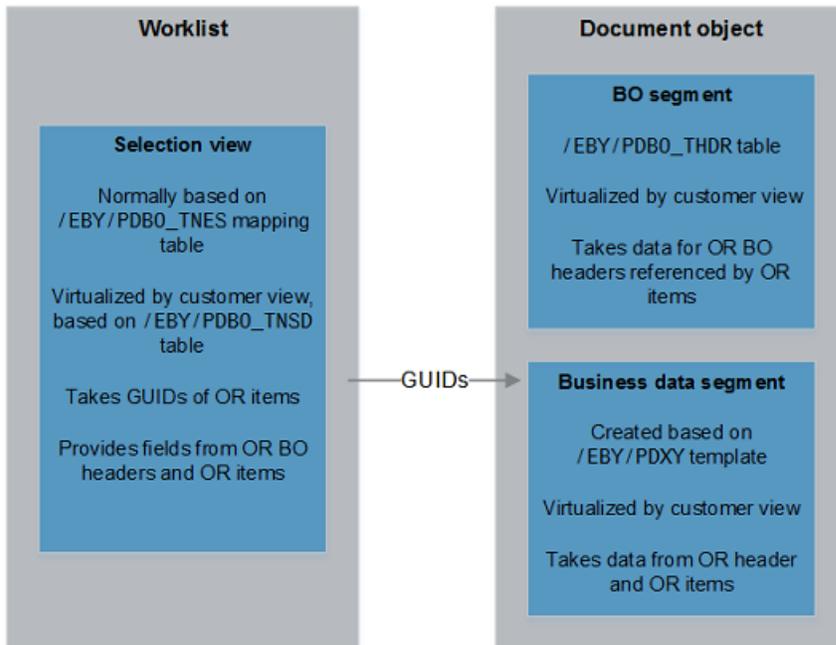
1. Select the **Delete object type** tab.
2. Enter an object type key and click the **Execute**  button to delete the object type's system configuration.

**Note** Any generated structures must be deleted manually.

## Display document item data in header

You can use a custom process type to display headers that consist of data from any document item. In the following example, the new process type is created based on the Generic Object Template (XY), and utilized to display the document item data.

The following image displays the corresponding setup.



In the data model, only the headers are defined. The actual header data is provided by a database view that can join any data. In this example, the view contains only the `/EBY/PDOR_TITM` table.

The data model interface can consist of data from this database view and from the Business Object (BO) interface structure, which will be automatically populated in the Data Access Object (DAO) from the BO database view, as depicted in the image.

The document objects do not exist in the `/EBY/PDBO_THDR` table. Therefore, the BO data has to be virtualized based on another database view. This view is provided to the DAO layer by the `/EBY/CL_PDBO_EVT_SET_DAO_PARAM` event.

From the fields in the grid, it is possible to jump to any related PD document.

## Configure the Generic Object Template (XY)

To configure the Generic Object Template, complete the following steps.

1. Create the new `ZPZ_PZOR2_VHDR` view that will be used as the data source.  
For typical document types, the view links to a database table. In this example, the view joins two tables.
2. In the new view, in the **Table/Join Conditions** tab, set up the following **Join conditions** for the `/EBY/PDOR_TITM` and `/EBY/PDOR_THDR` tables.

Table	Field name	=	Table	Field name
<code>/EBY/PDOR_TITM</code>	CLIENT	=	<code>/EBY/PDOR_THDR</code>	CLIENT
<code>/EBY/PDOR_TITM</code>	PARENTGUID	=	<code>/EBY/PDOR_THDR</code>	GUID

3. In the **View Fields** tab, set up the mandatory view fields as follows.

View field	Table	Field	Key	Data element
CLIENT	/EBY/PDOR_TITM	CLIENT		MANDT
GUID	/EBY/PDOR_TITM	GUID		OS_GUID

4. Create the ZPZ\_PZOR2\_LHDR table type.

To base it on the new view, in the **Line Type** tab, as the **Line Type**, select the new ZPZ\_PZOR2\_VHDR view.

5. Create the ZPZ\_PZOR2\_SHDRV structure.

In the **Components** tab, as the **Component Type**, include the new ZPZ\_PZOR2\_VHDR view and a version field.

Component	Typing method	Component type	Data type	Length	Decimal places	Short description
.INCLUDE	1 Types	ZPZ_PZOR2_VHDR		0	0	OR items and header
.INCLUDE	1 Types	/EBY/ PDBO_SPERSISTENCE_VERSION		0	0	Provides version fields for selection tables within the DAO

6. Build the ZPZ\_PZOR2\_LHDRV table type.

In the **Line Type** tab, as the **Line Type**, select the new ZPZ\_PZOR2\_SHDRV structure.

7. Use the dictionary objects that you created in the previous steps to run the PDXY wizard. The wizard will create the complete default configuration for the new ZPZORITEM2 document type.
  - a. In the expert IMG for the **XY Generic Object Template** object type, go to **Create new object type or add model level**.
  - b. In the **Create configuration of specific implementation of the generic PD obj.** screen, in the **Object type** box, enter the name of the process type for which you want to create the configuration. In this example: ZPZORITEM2.
  - c. Under **Required objects**, add the dictionary objects, as required.

**Important** Do not click the **Generate** button. The required objects have already been created in the previous steps.

- d. To create the configuration, click the **Execute** button.

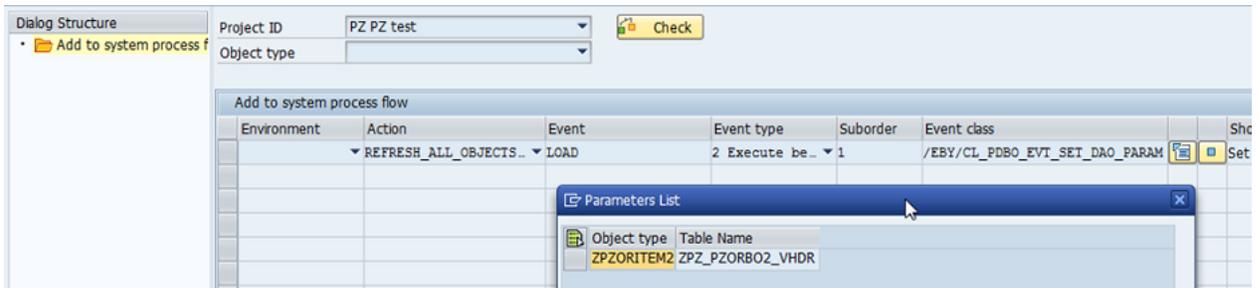
The result will look similar to this image.

Logical level	Parent level	Model data structure	DB table type	DB versions TTyp	Interf. struct.
HEADER		/EBY/PDOR_SITM_DATA	ZPZ_PZOR2_LHDR	ZPZ_PZOR2_LHDRV	ZPZ_PZOR2_SHDR_IF

## Set up the BO data virtualization

To set up the BO data virtualization, complete the following steps.

1. In the **Add to system process flow** dialog structure menu, define the view in the parameters of the /EBY/CL\_PDBO\_EVT\_SET\_DAO\_PARAM event, as follows.



The event is called in the REFRESH\_ALL\_OBJECTS action.

Setting	Entry
Environment	/
Action	REFRESH_ALL_OBJECTS
Event	LOAD
Event type	2 Execute before
Suborder 1	1
Event class	/EBY/CL_PDBO_EVT_SET_DAO_PARAM
Parameters	<ul style="list-style-type: none"> <li>• <b>Object type:</b> ZPZORITEM2</li> <li>• <b>Table Name:</b> ZPZ_PZORBO2_VHDR</li> </ul>

**Note** You can define views for more than one object type.

The view has to provide the CLIENT and GUID fields. In this example, the GUID field is taken from the OR item record in the /EBY/PDOR\_TITM table. The other entries are taken from the /EBY/PDBO\_THDR record of the OR document that this OR item belongs to.

2. In the new view, in the **Table/Join Conditions** tab, set up the following **Join conditions** for the /EBY/PDOR\_THDR and /EBY/PDOR\_TITM tables.

Table	Field name	=	Table	Field name
/EBY/PDOR_THDR	CLIENT	=	/EBY/PDOR_THDR	CLIENT
/EBY/PDOR_THDR	GUID	=	/EBY/PDOR_TITM	PARENTGUID

3. In the **View Fields** tab, set up the mandatory view fields as follows.

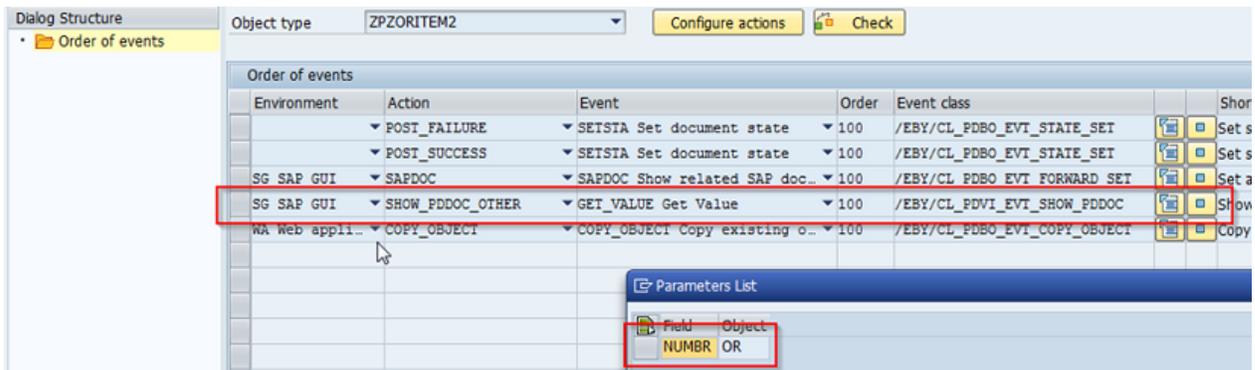
View field	Table	Field	Key	Data element
CLIENT	/EBY/PDBO_THDR	CLIENT		MANDT

View field	Table	Field	Key	Data element
GUID	/EBY/PDOR_TITM	GUID		OS_GUID

## Configure the jump to the Process Director document

To configure the jump to the Process Director document, complete the following steps.

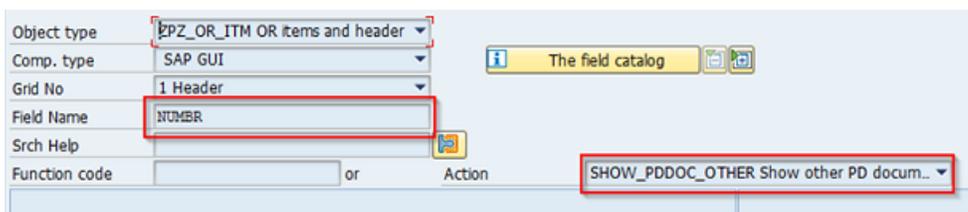
1. In the **Order of events** dialog structure menu, set up the /EBY/CL\_PDVI\_EVT\_SHOW\_PDDOC event that is used to process the document object.



The event is called in the SHOW\_PDDOC\_OTHER action. The destination document type is taken from the event parameters. The document number is taken from the source document field. The name of the field is also defined in the event parameters.

Setting	Entry
Environment	SG SAP GUI
Action	SHOW_PDDOC_OTHER
Event	GET_VALUE Get Value
Event type	2 Execute before
Order	100
Event class	/EBY/CL_PDVI_EVT_SHOW_PDDOC
Parameters	<ul style="list-style-type: none"> <li>• <b>Field:</b> NUMBR</li> <li>• <b>Object:</b> OR</li> </ul>

2. In the field catalog, in the view model, assign the SHOW\_PDDOC\_OTHER action to the NUMBR field. Leave the **Function code** field empty.



## Configure the Worklist

To configure the Worklist, complete the following step.

- Disable the Worklist root node. This node is static and will not work correctly because of the missing /EBY/PDBO\_THDR entry. Create semi-dynamic nodes below it.

## Nested set virtualization

The new document type does not have any records in the /EBY/PDBO\_TNES table. Therefore, this table has to be replaced with the /EBY/PDBO\_TNSD table, which already contains a default entry for an empty project. Create a new entry for your configuration project, using the following parameters.

Client: As required

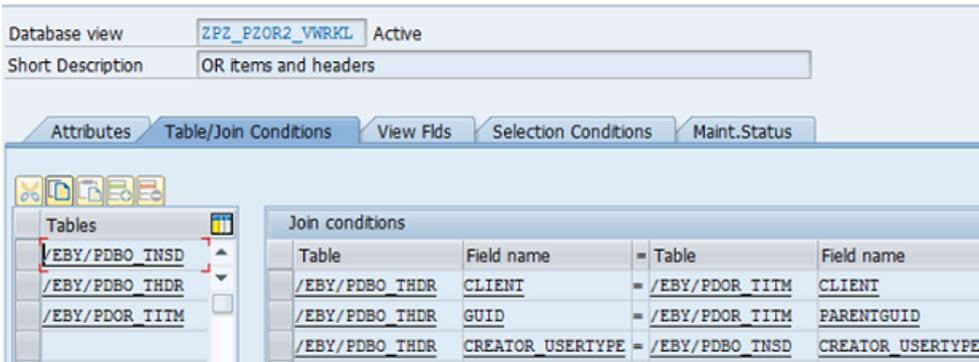
Project: As required

GUID: Empty

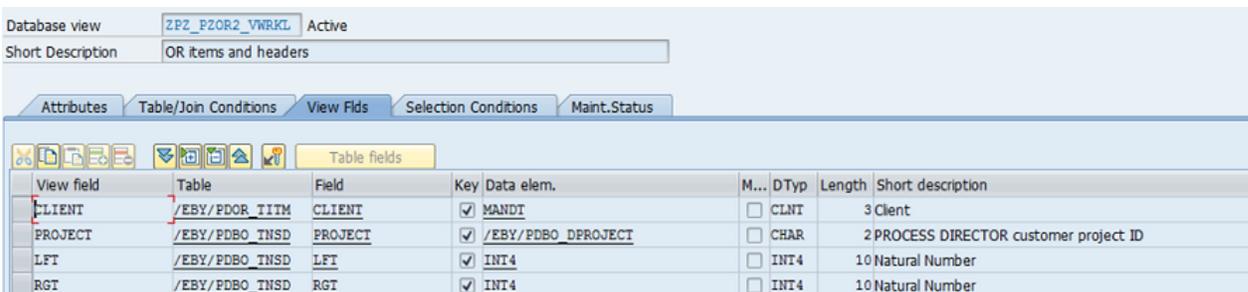
LFT: 9999

RGT: Empty

The following image displays an example of a possible Worklist view setup:



Like any other Worklist view, this view has to contain at least the following fields: CLIENT, PROJECT, GUID, LFT, and RGT. The PROJECT, LFT, and RGT fields have to be taken from the new /EBY/PDBO\_TNSD table.



## Field selection

In this example, the selection can be done based on any ITEM or BO field. It could also be done based on any OR header field.

**Display Worklist configuration / PZ test (PZ)**

View width Ranges editor mode Start-up selection screen Selection screen

Structure	Obj...	Auth. object	Show ...	View name	Worklist node type
PO	PO			ZPZ_PDPO_VWRKLS	/EBY/CL_PDBO_WRK
IV	IV			/EBY/ICIV_VWRKLS	Static
PA	PA			ZEBY_PDPA_VWRKLS	Static
PR	PR			ZPZ_PDPR_VWRKLS	/EBY/CL_PDPR_WOR
MG_GLACC	MD_...				Static
DN	DN			ZPZ_PDDN_VWRKLS	Static
WC	WC			ZPZ_PDWC_VWRKLS	/EBY/CL_PDWC_WOR
MD_VENDOMA	MD_...				Static
SO	SO			/EBY/PDSO_VWRKLS	Static
OR	OR				Static
AR	AR				Static
AT	AT	ZEBY_FI			Static
ES	ES				Static
IB	IB				Static
FIC	FIC				Static
FI	FI				Static
ZPZ_AR	ZPZ_...				Static
MD_CUSTMA	MD_...				Static
ZPZORITEMS	ZPZ_...				Static
<b>ZPZORITEM2</b>	<b>ZPZ_...</b>			<b>ZPZ_PZOR2_VWRKLS</b>	<b>Static</b>
• All	ZPZ_...			ZPZ_PZOR2_VWRKLS	Semi-dynamic
• EUR	ZPZ_...			ZPZ_PZOR2_VWRKLS	Semi-dynamic
• USD	ZPZ_...			ZPZ_PZOR2_VWRKLS	Semi-dynamic

## Example

The following image displays a sample implementation of the new ZPZORITEM2 process type.

**Display PZ OR Items 2 #8 (11 of 15)**

Worklist	Total	Header	Item Material	Manufacturer Part No.	Vendor Material Number	UoM	Delivery Date	Quantity	Net Order Price	Crcy	Net Value	Gross value	Purch.Doc.	DocCh.	Ship date
PO	10/247	16	10			PC	21.05.2015	1	3,000.00	EUR	0.00	0.00	4500017...		
IV	10/1,722		20			PC	21.05.2015	1	0.00	EUR	0.00	0.00	4500017...		
PA	10/736	15	10			PC	21.05.2015	1	0.00	EUR	0.00	0.00	4500017...		
PR	10/30		20			PC	21.05.2015	1	0.00	EUR	0.00	0.00	4500017...		
MG_GLACC	4/4		13						0.00	EUR	0.00	0.00	4500017...		
DN	10/59		50						0.00	EUR	0.00	0.00	4500017...		
WC	10/1,259		200						0.00	EUR	0.00	0.00	4500017...		
MD_VENDOMA	10/27		30			PC	25.09.2014	1	0.00	EUR	0.00	0.00	4500017...		
SO	10/164	9	100-100			PC	08.09.2014	3	0.00	EUR	0.00	0.00	4500017...		08.09.2...
OR	10/16		40	100-200		PC	08.09.2014	4	0.00	EUR	0.00	0.00	4500017...		08.09.2...
AR	2/2	8	20			PC	08.09.2014	3	0.00	EUR	0.00	0.00	4500017...		08.09.2...
ES	10/142		40	100-200		PC	08.09.2014	4	0.00	EUR	0.00	0.00	4500017...		08.09.2...
IB	10/98		4	20		PC	22.04.2014	1	0.00	EUR	0.00	0.00	4500017...		
FIC	10/18		2			PC	25.09.2014	2	0.00	EUR	0.00	0.00	4500017...		
FI	10/214		3			PC	22.04.2014	1	0.00	EUR	0.00	0.00	4500017...		
ZPZ_AR	4/4														
MD_CUSTMA	10/11														
MD_CUSTMA	10/11														
ZPZORITEMS	4/4														
<b>ZPZORITEM2</b>															
• All	21/21														
• EUR	15/15														
• USD	3/3														

## Settings

/EBY/PDXY\_COPY

### Create new object type

Setting	Description
Request/Task	Use search help to pick an existing request or create a new one.
Source type of configuration copy	Unless you want to copy from an existing object type, you should set this to: XY Generic Object Template.
Object type	Enter a unique 2-10 character alphanumeric key. It must start with Y or Z.
Description	The description of the object type to be created.
Name pattern	The name pattern assists you in entering names for the required objects. Enter a pattern and click the <b>Align</b>  button. Names based on this pattern are entered in the Model data structure field and in all fields in the Required objects section.
Model data str.	Enter the name for a structure containing the fields available on this document model level which should be persisted to the database. Important: This structure must already exist. For the document model level HEADER, you should not include any BO fields in the model data structure, as these will be automatically included in the interface structure. For new object types, the default document model level is HEADER.
Table name	Enter a unique name for a database table to hold header records. The name must follow the customer's development naming conventions. This object will be generated when you click the Generate button, and will have the model data structure as its line type.
DB table type	Enter a unique name for the database table's table type. The name must follow the customer's development naming conventions. This object will be generated when you click the Generate button.
Version struct.	Enter a unique name for the version structure. The name must follow the customer's development naming conventions. This object will be generated when you click the Generate button, and will include the DB table type plus a version field.

Setting	Description
DB version TTyp	Enter a unique name for the version structure table type. The name must follow the customer's development naming conventions. This object will be generated when you click the Generate button.
Interf. struct	Enter a name for a structure with which the model can expose its own fields. The name must follow the customer's development naming conventions. This object will be generated when you click the Generate button and will include the header model data structure plus the administrative fields from the base object.

## Add model level

Setting	Description
Request/Task	Use search help to pick an existing request or create a new one.
Object type	Select an object type to which you want to add model levels from the drop-down list.
Parent level of the internal data model	Select a document model level from the drop-down list to serve as the parent level for the model level you want to add.
Logical level of the internal data model	Enter an alphanumeric name for the new logical level, or use search help to pick an existing level (the level's object type must match the one selected above).
Name pattern	The name pattern assists you in entering names for the required objects. Enter a pattern and click the <b>Align</b>  button. Names based on this pattern are entered in the Model data structure field and in all fields in the Required objects section.
Model data str.	Enter the name for a structure containing the fields available on this document model level which should be persisted to the database. This structure must already exist. Note that this structure is also used as the interface structure, as no administrative fields need to be exposed.
Table name	Enter a unique name for a database table to hold records of the given model level. The name must follow the customer's development naming conventions. This object will be generated when you click the <b>Generate</b> button, and will have the model data structure as its line type.
DB table type	Enter a unique name for the database table's table type. The name must follow the customer's development naming conventions. This object will be generated when you click the <b>Generate</b> button.

Setting	Description
Version struct.	Enter a unique name for the version structure. The name must follow the customer's development naming conventions. This object will be generated when you click the <b>Generate</b> button, and will include the DB table type plus a version field.
DB version TTyp	Enter a unique name for the version structure table type. The name must follow the customer's development naming conventions. This object will be generated when you click the <b>Generate</b> button.

### Example

**Create configuration of specific implementation of the generic PD obj.**

Request/Task: N47K910024

Source type of configuration copy: XY Generic Object Template  
 Object type: ZTRAVEL  
 Description: Travel request 2

**Data management**

Name pattern: ZEBY\_PDZT2  
 Model data str.: ZEBY\_PDZT2\_SHDR\_DATA

**Required objects**

Table name	ZEBY_PDZT2_THDR	
DB table type	ZEBY_PDZT2_LHDR	
Version struct.	ZEBY_PDZT2_SHDRV	
DB version TTyp	ZEBY_PDZT2_LHDRV	
Interf. struct.	ZEBY_PDZT2_SHDR_IF	

**Next steps**

- Review number range and SAP business object settings
- Review field display properties
- Display IMG structure

# Configure quantity conversion rules

## What are quantity conversion rules?

Process Director can be configured to check the quantity in an order confirmation against the quantity in the related purchase order. This check is done using the base unit of measure.

Quantity conversion rules enhance this check with alternative units of measure, which are defined in relation to the base unit of measure. Example: A conversion rule is set up that one box (alternative unit of measure) contains 10 pieces (base unit of measure). A delivery of one box would then successfully pass the check, as would a delivery of 10 pieces.

Quantity conversion rules are also applied to order confirmations that are related to a scheduling agreement. In addition, these rules are applied to order confirmations that are posted using the Web Application too.

**Note** Quantity conversion rules for purchase orders are taken into account only if the Check confirmed quantity check is activated.

## Configure a quantity conversion rule

To configure a quantity conversion rule, complete the following steps.

1. In the **MM02** SAP transaction, enter the material number for which you want to configure a conversion rule.
2. From the **Select View(s)** dialog box, select the **Basic Data 1** view.
3. On the Application Toolbar, click **Additional Data**.
4. On the **Additional EANs** tab, define the alternative unit of measure that you want to use.
5. On the **Units of measure** tab, configure the conversion rule.  
Example: For one box to contain 10 pieces, set **BOX** as the alternative unit of measure, **10** as the numerator for the conversion, and **PC** as the base unit of measure.
6. Click the **Save**  button.

# Integrate with SAP Business Workflow

## About SAP Business Workflow integration

You can integrate Process Director workflows in SAP Business Workflow. If a user sends a Process Director workflow step to an SAP user, it also appears as a new work item in the Workflows category of the recipient's SAP Business Workplace Inbox in transaction `SBWP`.

To integrate with SAP Business Workflow, complete the following procedures in transaction `SWDD`.

1. [Create a new workflow definition](#)
2. [Create a container element](#)
3. [Set up the workflow start event](#)
4. [Add a container operation to delete superfluous attachments](#)
5. [Add an activity for the Process Director workflow step.](#)
6. [Activate the workflow definition](#)

## Create a new workflow definition

The first step in integrating Process Director in SAP Business Workflow is to create a new workflow definition.

To create a new workflow definition, complete the following steps.

1. In transaction `SWDD`, in the Workflow menu, click **New**, then save.
2. In the **Give your new workflow template a name here** dialog box, type an abbreviation and a name and click the **Continue**  button.
3. In the **Create Object Directory Entry** dialog box, select a **Package**, then save.
4. Next, [create a container element](#).

## Create a container element

A container element maps a Process Director workflow step to an SAP Business Workflow step.

To create a container element, complete the following steps.

Prerequisite: You have created a workflow definition.

1. In transaction `SWDD`, below the **Navigation Area**, select **Workflow Container** tab and double-click **<Double-Click to Create**.

2. In the **Create Container Element** dialog box, type an element, a name, a description and the object type.
3. On the **Properties** tab, select the **Parameter settings** > **Import** check box.
4. Click the **Continue**  button.
5. Next, [set up the workflow start event](#).

## Set up the workflow start event

The workflow start event starts an SAP Business Workflow when a Process Director workflow step starts.

To set up the workflow start event, complete the following steps.

Prerequisite: You have created a container element.

1. In transaction `SWDD`, click the **Basic data**  button.
2. On the **Version-Independent (Task)** > **Start Events** tab, add the following entries, then press **Enter**.

Field	Setting
Object Category	BO BOR Object Type
Object Type	
Event of the object	STARTED

3. Click the **Activate**  button to activate the event linkage. The system prompts you to enter a customizing request.

**Note** In some customer implementations, you may have to activate the event linkage manually in every client and system.

4. Click the **Binding**  button to configure the event binding.
5. In the **Change Binding for Workflow** dialog box, drag and drop the event container field `_EVT_OBJECT` onto the `SingleDocumentLink` entry of your new workflow container.  
If an event is raised, the instance of the object that raised the event is available in the event container field `_EVT_OBJECT`. This field must be copied into the corresponding field in the workflow container. This is the field `SingleDocumentLink`, which was created in the workflow step. It is the only available field for event binding, because it is the only field marked as an input field.
6. Click the **Continue**  button.
7. To specify how the work item recipients will be determined, complete the following substeps.
  - a. Click the **Agent Assignment for task**  button.
  - b. In the **Maintain Agent Assignment** dialog box, click the **Attributes** button.
  - c. Select the **General task** option and click the **Transfer** button. This setting allows all SAP users as possible work item recipients.
  - d. Click the **Update Index**  button to activate the assignment.

Next, add a [container operation](#) to delete superfluous attachments.

## Add a container operation

Any actual step processing occurs in Process Director, so only the work item itself should display in SAP Business Workflow. You must therefore add a container operation to remove any superfluous attachments that might be confusing. Because you do not need to transport any attachments from Process Director into *SBWP*, you can simply delete them from the workflow. When the user selects the work item, Process Director opens.

To add a container operation, complete the following steps.

Prerequisite: You have set up the workflow start event.

1. Click the **Back**  button to return to the overview.
2. In the workflow visualization, double-click **Undefined- Double-click**.
3. In the **Step Selection** dialog box, double-click the **Container Operation** step type.
4. In the **Control** tab, type the following settings.

Field	Setting
Step name	Delete useless attachments
Outcome name	Attachments deleted
Step not in workflow log	Select the check box.
Result element	_Attach_Objects

5. Click the **Continue**  button.

Next, [add an activity for the Process Director workflow step](#).

## Add an activity for the Process Director workflow step

To add an activity for the Process Director workflow step, complete the following steps.

Prerequisite: You have added a container operation.

1. Right-click **Workflow completed**, then select the context menu item **Create**.
2. In the **Step Selection** dialog box, double-click the **Activity** step type.
3. In the **Control** tab, in the **Task** dropdown menu, select **Create task**.
4. In the **Task creation** dialog box, in the **Basic Data** tab, type the following settings.

Field	Setting
<b>Abbr</b>	PDWCProcess
<b>Name</b>	Process Director workflow process
<b>Object Category</b>	BO BOR Object Type
<b>Object Type</b>	/EBY/PDWC
<b>Method</b>	DISPLAY

5. Optional. In the **Description** tab, add documentation for the task.
6. In the **Terminating events** tab, define a terminating event with the following settings.

Field	Setting
Element	_WI_OBJECT_ID
<b>Object Category</b>	BO BOR Object Type
Object Type	/EBY/PDWC
Event	FINISHED

This ensures that the work item disappears from the SAP Business Workplace if a user cancels, recalls or finishes it from within Process Director.

7. Save your settings.
8. In the **Create Object Directory Entry** dialog box, type an appropriate Z package.

**Important Important:** Check that everything is correct before returning to the activity definition. You will no longer be able to edit the task after returning to the activity definition.

9. Click the **Back**  button to return to the activity definition dialog box.
10. The system displays a proposal for a binding. Click the **Continue**  button to confirm.
11. In the **Control** tab, in the **Agents** section, use the search help to select an expression for the **Agents** assignment.
12. In the **Expression for step agent** dialog box, select the expression **Container** > **SingleDocumentLink** > **User** .
13. Click the **Continue**  button.
14. In the **Control** tab, in the **Task Properties** section, click the **Agent assignment for task**  button.
15. In the **Maintain Agent Assignment** dialog box, click the **Attributes** button.
16. In the **Task** dialog box, select the **General Task** option and confirm.
17. Click the **Back**  button twice to return to the Workflow Builder.
18. Next, [activate the workflow definition](#)

## Activate the workflow definition

Activating the workflow definition finalizes the configuration and takes it into production.

To activate the workflow definition, complete the following step.

Prerequisite: You have added an activity for the Process Director workflow step.

- Click the **Activate**  button.