PROCESS DIRECTOR 7.3

Reference Guide

September 2016 – ReadSoft AB

The contents of this document are subject to change without notice.

Questions or comments about this document may be emailed to **documentation@readsoft.com**.

02092016.1236

ReadSoft AB (Head office) | Södra Kyrkogatan 4 | SE-252 23 Helsingborg | Sweden | Phone: +46 42 490 21 00 | Fax: +46 42 490 21 20
ReadSoft AG | Falkstrasse 5 | 60487 Frankfurt | Germany | Phone: +49 69 1539402-0 | Fax: +49 69 1539402-13
info@readsoft.com | www.readsoft.com

Contents

Introduction

About this guide

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

The guide provides reference information for PROCESS DIRECTOR.

For information on installing and configuring PROCESS DIRECTOR, see the following guides:

- *Importing PROCESS DIRECTOR into SAP*
- *PROCESS DIRECTOR SAP Configuration Guide*
- *PROCESS DIRECTOR SAP Advanced Configuration Guide*
- *PROCESS DIRECTOR Troubleshooting Guide*

Typographical conventions

The following typographical conventions are used:

- Menu names, commands, and dialog names appear in **bold**.
- Names of keys on your keyboard appear in SMALL CAPITAL LETTERS.
- Names of files, folders, and settings appear in `Courier` font.
- SAP transaction codes and program names appear in `ARIAL MONOSPACE`.
- Variables that have to be replaced by an actual value are *italicized* or appear in <>.

General reference

This section provides reference information that applies to all process types. It covers the following subjects:

- Query classes
- Mapping conversions
- Preset IDs
- Preset classes
- Checks
- Determinations
- Mechanisms for populating fields
- Workflow control patterns and step types
- Standard documentation objects
- Placeholders for documentation objects
- User profile parameters
- Programs

See the Process types reference section for process type specific reference information.

Available query classes

Query classes are used in the ranges configuration of semi-dynamic and dynamic Worklist nodes to supply parts of the query at runtime. For example, the query class `/EBY/CL_PDBO_WL_QRY_PD_USER` determines the name of the currently logged on user and displays all documents assigned to that user in the Worklist node.

The following query classes are available in the PROCESS DIRECTOR standard:

Query class name	Description
/EBY/CL_PDBO_WL_QRY_ALPHA_CONV	**Alpha conversion** To convert the query result from internal to external format. This query class is useful for dynamic nodes, where the field content forms the node name and therefore may not be easily readable. For example, if you build dynamic nodes for individual vendors, without alpha conversion the node names would be something like 0000001000, 0000002000, etc. If you use the alpha conversion query, the node descriptions would be 1000, 2000, etc.
/EBY/CL_PDBO_WL_QRY_CUR_DAY	**Current day (sy-datum) +/- days** To include documents whose date is within a specific time frame, for example, documents from the last week or the last month.
/EBY/CL_PDBO_WL_QRY_CUR_DAY_TS	**Time stamp of the current day +/- days** To include documents whose timestamp is within a specific time frame.
/EBY/CL_PDBO_WL_QRY_PD_SUB_ACT	**Active substitutes query** To include documents of active and passive substitutes. **Note**: When you use this query, the fields WC_PROCESSER and WC_PROC_TYPE must be included in the node's view.
/EBY/CL_PDBO_WL_QRY_PD_SUBST	**PD user substitutions** To include documents of active substitutes only. **Note**: When you use this query, the fields WC_PROCESSER and WC_PROC_TYPE must be included in the node's view.

Query class name	Description
/EBY/CL_PDBO_WL_QRY_PD_USER	**PD user name** To include documents of the currently logged on user.
/EBY/CL_PDBO_WL_QRY_PD_USER_TY	**PD user type** To include documents of the currently logged on user type: S - SAP user I - Internet user L - LDAP user
/EBY/CL_PDBO_WL_QRY_STATE_DESC	**State description** To create dynamic nodes for individual document statuses and display status descriptions (**New**, **Processed**, etc.) instead of the status value (**NN**, **OK**, etc.).

Available mapping conversions

Mapping conversions enable you to perform functions on fields when they are imported into PROCESS DIRECTOR. These mapping conversions are available in the PROCESS DIRECTOR standard:

Mapping function module	Description
/EBY/PDBO_MAF_ADD	**Add parameter to value** For numeric fields, adds the value specified in the parameters to the field value. **Example**: An original field value of 1000 and a parameter value of 100 results in a value of 1100 after conversion.

Mapping function module	Description
/EBY/PDBO_MAF_ADD_TEXT	**Write input to a text defined by text ID** Saves an external string to a PROCESS DIRECTOR text. 1. Map the external field containing the string to any internal field of the appropriate PROCESS DIRECTOR logical level (HEADER or ITEMS). Note that this internal field will not be filled. 2. Create a mapping conversion using the /EBY/PDBO_MAF_ADD_TEXT function. In the parameters, specify the text type and ID, for example, EKKO for purchasing document header texts, or /EBY/PD for PROCESS DIRECTOR notes. You can use any of the defined supported text types. 3. Assign the mapping conversion to the external field.
/EBY/PDBO_MAF_ARITHMETIC	**Process arithmetic functions on input** Performs the arithmetic function defined in the parameters on the field value. **Example**: Use the TRUNC function to remove decimal places, for example, to truncate 1.4142 to 1. (Note that this function performs no rounding, so 1.999 is also truncated to 1). See the SAP documentation for detailed information on the available arithmetic functions.

Mapping function module	Description

`/EBY/PDBO_MAF_ASSIGN`	**Assign a variable to output**
	Assigns an SAP system variable or other interface variable to the output. Enter the variable in the parameters.
	The following interface variables are currently supported:
	`IR_MAPPER`
	`IS_CURRENT_DMO`
	`IS_CURRENT_MAP`
	`IN_SUBNODE`
	`IN_PARENTSUBNODE`
	`IR_CURRENT_PARENT`
	`IR_CURRENT_OBJECT`
	Example: Enter `IN_SUBNODE` in the parameters to sequentially number the line items: 1, 2, 3.
	Tip: To number the generated line item numbers in a 10, 20, 30 sequence instead of 1, 2, 3, additionally enter the `/EBY/PDBO_MAF_MULTIPLY` mapping function with a parameter of 10, followed by the `/EBY/PDBO_MAF_ARITHMETIC` mapping function with the parameter `TRUNC`.

Mapping function module	Description
`/EBY/PDBO_MAF_CLEAR`	**Initialize the value** Removes the value from the field. Generally speaking, if you do not want a value in a specific field in the PROCESS DIRECTOR document, you should not add the field to the mapping table. However, there may be cases where you want to capture a value, but remove the value if it does not meet certain conditions. **Example**: Your PROCESS DIRECTOR field is limited to 5 characters, while the captured field may contain up to 20 characters. In this case, only the first 5 characters of the captured value are inserted in the field, which may not make sense, so you may want to clear these characters from the output field and set the field value later in the PROCESS DIRECTOR document. To achieve this, first add the mapping conversion **Skip next function unless condition is met** (`/EBY/PDBO_MAF_IF_LEN`) and in the parameters, select the operator `GT` and enter 5 in the next field (this specifies the condition "field contains more than 5 characters"). Assign the order 1 to this conversion. Next, add the mapping conversion **Initialize the value** and assign the order 2 to this conversion. Result: The mapping conversion **Initialize the value** is skipped if the captured field contains 5 characters or less.
`/EBY/PDBO_MAF_DATEADJUST`	**Adjust date (add/subtract working days)** Changes the date by adding or subtracting the number of working days specified in the parameters.

Mapping function module	Description
/EBY/PDBO_MAF_DATECONVERT	**Check/convert date to internal format** Checks whether a date is in SAP internal format or converts the date to SAP internal format. **Example**: A date in the format MM/DD/YYYY (for example, 01/23/2012) is converted to the internal format YYYYMMDD (20120123).
/EBY/PDBO_MAF_DEL1ST	**Delete first occurrence of parameter from output** Deletes the first occurrence of the string specified in the parameters from the output. **Example**: A captured order number contains two leading zeroes, which should not be present in the PROCESS DIRECTOR field. Enter 00 in the parameters. 00 at the beginning of the field is removed by the mapping conversion, but 00 within the field is retained.
/EBY/PDBO_MAF_DELETE	**Delete all occurrences of a parameter from input** Replaces all occurrences of the string specified in the parameters.
/EBY/PDBO_MAF_DIVIDE	**Divide input with parameter** For numeric fields, divides the field value by the value specified in the parameters. **Example**: An original field value of 1000 and a parameter value of 100 results in a value of 10 after conversion.

Mapping function module	Description
`/EBY/PDBO_MAF_FILTER_ALLOWED`	**Remove all characters which do not appear in parameter list** Removes all characters that are not specified in the parameter list from the field. **Example**: Enter 0123456789 in the parameters to remove non-numerical characters such as . / - from date fields.
`/EBY/PDBO_MAF_FILTER_FORBIDDEN`	**Remove all characters which do appear in parameter list** Removes all characters specified in the parameter list from the field. **Example**: Enter characters such as - _ ( ) [ ] in the parameters to remove these characters from telephone numbers.
`/EBY/PDBO_MAF_FIXEDVALUE`	**Use fixed value from parameters** Inserts the value specified in the parameters into the field. This is useful for entering values into fields that are not captured during mapping. It can be used as an alternative to presets. **Example**: If your Customer Orders are always of the type `OR Standard Order`, you can automatically populate the **Sales document type** field with the value `OR` by entering this value in the parameters.

Mapping function module	Description
/EBY/PDBO_MAF_IF	**Skip next function unless condition is met** Skips the next mapping conversion unless the condition specified in the parameters is met. You specify the sequence in which mapping conversions are performed by entering the appropriate number in the **Order** column of the mapping conversion configuration. This mapping conversion is useful, for example, for determining whether a value is already in a field and, if it is not, entering a predefined value. **Example**: Suppose you want to ensure that a date field is filled. If the user enters a date, this value is not changed; if the user does not enter a date, the current system date is inserted. To achieve this, first add the mapping conversion **Skip next function unless condition is met** (/EBY/_PDBO_MAF_IF) and in the parameters, select the option EQ and leave the **Text** field blank (this specifies the condition "field = empty"). Assign the order 1 to this conversion. Next, add the mapping conversion **Get a system value** and enter DATLO in the parameters (this adds the system date to the field). Assign the order 2 to this conversion. Result: The mapping conversion **Get a system value** is skipped if the field is filled by the user.
/EBY/PDBO_MAF_IF_LEN	**Skip next function unless condition is met** Skips the next mapping conversion unless the condition specified in the parameters is met. This function checks only the field length. To check field contents, use /EBY/PDBO_MAF_IF. **Example**: To skip the next function if the field length is less than 10, enter the operator GT and the value 10 in the parameters.

Mapping function module	Description
/EBY/PDBO_MAF_LOWERCASE	**Change the value to lower case** Changes the value of the field to lower case characters. **Example**: Changes the field value TEXT or Text to text.
/EBY/PDBO_MAF_MULTIPLY	**Multiply input with parameter** For numeric fields, multiplies the field value by the value specified in the parameters. **Example**: An original field value of 1000 and a parameter value of 100 results in a value of 100,000 after conversion.
/EBY/PDBO_MAF_NOSPACE	**Change the value by removing all space characters** Removes all spaces from the field value. **Example**: Converts the telephone number 001 555 21 21 to 0015552121.
/EBY/PDBO_MAF_OTHER_FIELD	**Map different field from the same level** In the parameters of this mapping function you can specify an external field from the same data level as the field that is to be mapped. Using this approach, you can fill the field with the value from a different field if the original field is empty.
/EBY/PDBO_MAF_POSTFIX	**Append parameter to value** Adds the string specified in the parameters at the end of the field value.
/EBY/PDBO_MAF_PREFIX	**Prepend parameter to value** Adds the string specified in the parameters at the beginning of the field value.

Mapping function module	Description
/EBY/PDBO_MAF_REPLACE	**Replace all occurrences of parameter A with B** Replaces all occurrences of the first specified parameter with the second specified parameter. **Example**: Replaces all commas (first parameter) with a decimal point (second parameter).
/EBY/PDBO_MAF_SAMPLE	**Sample mapping function** This function is a template that you can use to create your own mapping functions. **Note**: Do not assign this mapping function to a conversion. Instead, copy the template to create your own function and assign that function to the conversion.
/EBY/PDBO_MAF_SPACE_BUC	**Insert space before first upper case character** Inserts a space before the first upper case character found in the field.
/EBY/PDBO_MAF_SUBTRACT	**Subtract parameter from value** For numeric fields, subtracts the value specified in the parameters from the field value. **Example**: An original field value of 1000 and a parameter value of 100 results in a value of 900 after conversion.
/EBY/PDBO_MAF_SYST	**Get a system value** Enters the system value specified in the parameter list in the field. **Example**: Enters the current system date when DATLO is entered in the parameter list.

Mapping function module	Description
`/EBY/PDBO_MAF_UOM`	**Unit of measurement ISO or string to SAP** Converts an ISO unit of measurement or a text string to the SAP internal format required by PROCESS DIRECTOR. **Example**: Converts the text string `Piece(s)` or the ISO code `PCE` to the SAP value `PC`. **Note**: This mapping conversion uses a number of checks to find the correct SAP format. If all checks fail, the field is cleared.
`/EBY/PDBO_MAF_UPPERCASE`	**Change the value to upper case** Changes the value of the field to upper case characters. **Example**: Changes the field value `text` or `Text` to `TEXT`.

Available preset IDs

A preset ID determines at what point in the life cycle of a document the preset value is applied. These preset IDs are available in the PROCESS DIRECTOR standard and should usually be enough for most usage scenarios:

Preset ID	Applies presets
`APPROVE`	When a document is approved in a workflow step.
`CHECK`	Before checks and determinations run.
`CREATE`	When a new document or a new line item or other sub-object (accounts, partners, etc.) is created.
`INSERT`	When a newly created document is saved for the first time (that is, inserted in the database).
`POST_SUCCESS`	When a document is successfully posted.

Preset ID	Applies presets
RECALL	When a workflow is recalled. **Note**: This applies only to workflow recall; it does not apply to the removal of individual processors from a workflow step.
REJECT	When a document is rejected in a workflow step.
UPDATE	When a document is saved.

Available preset classes

A preset class enables you to dynamically compute a value which can then be applied to a field as a preset. For example, a preset class can determine the company code entered in a document and populate the **Currency** field with the currency defined for that company code.

The following general preset classes are available for all process types in the PROCESS DIRECTOR standard:

Preset class name	Description
/EBY/CL_PDBO_PRESET_CURR_AMNT	**Convert a currency amount to SAP internal format** Uses the field CURRENCY from the same level or the first field of type WAERS to identify the currency and converts the input value to SAP internal format. This preset must be used for all currency value fields which are imported using the preset ID CREATE.
/EBY/CL_PDBO_PRESET_CURR_OF_CC	**Preset currency from company code**

Preset class name	Description
/EBY/CL_PDBO_PRESET_TEMPL	**Preset template** This preset class is provided as a template that you can use to configure your own custom presets. **Note**: Do not add this preset class to the presets configuration. Instead, create your own preset class based on the template and add this custom preset class to the configuration.

See the following sections for information on process type specific preset classes:

Preset classes for Asset Retirement

Preset classes for Asset Transfer

Preset classes for Electronic Bank Statements

Preset classes for Financial Postings

Preset classes for Goods Receipts

Preset classes for Master Data Maintenance

Preset classes for Payment Advices

Preset classes for Requisitions

Available checks

A check verifies the completeness, accuracy and consistency of data in a business document.

The following general checks are available for all process types in the PROCESS DIRECTOR standard.

Check event name	Description
/EBY/CL_PDBO_EVT_DFC_CHK	**Duplicate value check** Checks if a defined field value or combination of field values entered by the user is already available in the database. This prevents duplicate records being added to the database. You define the field values that should be checked in the duplicates configuration.
/EBY/CL_PDBO_EVT_CHK_ATTACHMNT	**Check if the requested attachments exist** Checks if the document has an attachment in the format specified in the initialization parameters (for example, a PDF file). Add this check if users are required to add an attachment to the document.
/EBY/CL_PDBO_EVT_CHK_TEMPL	**Check template** This check is provided as a template that you can use to configure your own custom checks. **Note**: Do not add this check to the checks configuration. Instead, create your own check using the template and add this check to the configuration.
/EBY/CL_PDBO_EVT_APE_CHK	**Check if document is excluded from autoposting** For document-driven process types, checks whether an error-free document can be automatically posted or not depending on specific field values. You specify the field values in the check's initialization parameters. **Note**: You can also define the field values that should be checked in the autoposting configuration.

Check event name	Description
/EBY/CL_PDWC_EVT_CHK_FLDST_MND	**Mandatory field check based on field status** Checks the mandatory fields from the field status and returns messages if these fields have not been filled.

See the following sections for information on process type specific checks:

Check for Asset Acquisition

Checks for Asset Retirement

Check for Asset Transfer

Checks for Customer Orders

Checks for Financial Postings

Check for Goods Receipts

Checks for Invoice Block and Cancelation

Checks for Master Data Maintenance

Checks for Order Confirmations

Checks for Payment Advices

Checks for Payment Approvals

Checks for Requisitions

Available determinations

A determination attempts to infer the correct value for a given document field from the other fields in the document, or from other sources such as mapping tables.

There are no determinations that apply for all process types. See the following sections for information on the available determinations per process type in the PROCESS DIRECTOR standard:

Determinations for Customer Orders

Determinations for Goods Receipts

Determinations for Order Confirmations

Determinations for Payment Advices

Determinations for Requisitions

Available process parameters

PROCESS DIRECTOR provides an easy way for you to control the behavior of the system for the following processes:

- Archiving
- Checks
- Posting
- Workflow
- ... and more

No knowledge of PROCESS DIRECTOR actions and events (which control these processes) is required; all you have to do is set the appropriate parameters for the process.

See the *PROCESS DIRECTOR SAP Configuration Guide* for detailed information on configuring process parameters.

The following general parameters are available for all process types. See the process types reference section for information on process parameters for specific process types.

Archiving

`/EBY/PDBO_EPC_ARCHIV`

Initial settings > Process parameters

Process	Description
Add only the current workflow to the Smart Form	In the parameters, activate **Process current workflow only** to include only the details of the current workflow in the workflow log. If this option is not activated, details of all workflows to which the document was previously sent, as well as the current workflow, are listed in the workflow log.

Process	Description
Archiving object parameters	Automatically creates a log of all activities that took place on a document during a workflow and adds this as an attachment to the PROCESS DIRECTOR document and the corresponding SAP document. In the parameters, enter the archiving document type and the name of the Smart Form that should be used to create the workflow log. Select a status and/or substatus to create the workflow log only when the document has this status/substatus. See also Workflow log creation.
Archiving object type assignment to MIME type	
Archive settings for EDI data	Automatically creates an attachment that visualizes documents received via EDI (similar to the images of scanned documents that may be used with other process types). In the parameters, enter the archiving document type and the name of the Smart Form that should be used to create the attachment. See also Configuring IDoc image attachments.
Configure attachment deletion in WebApp	Specifies whether the archive document itself, or only the link, should be removed when users delete attachments in the Web Application. **Note**: Do not use options 0 and 1, they are not applicable.
Cover sheet properties	Automatically creates a cover sheet for request-driven documents when these are saved in the Web Application. In the parameters, enter the archiving document type and the name of the Smart Form that should be used to create the cover sheet. See also Configuring cover sheets.
Exclude document types for upload	In the parameters, add document types to prevent Web Application users from adding these document types as attachments to PROCESS DIRECTOR documents.

Process	Description
Smart Form and document type for archiving after posting	Automatically adds the workflow log as an attachment to the PROCESS DIRECTOR document and the corresponding SAP document when the document is posted. In the parameters, enter the archiving document type and the name of the Smart Form that should be used to create the workflow log. Select a status and/or substatus to create the workflow log only when the document has this status/substatus.
Wait x seconds for audit log archiving to finish	Enables you to delay archiving for a specified number of seconds to ensure that the PROCESS DIRECTOR workflow log and notes are archived correctly and available in the corresponding SAP documents. Enter a value here if you are experiencing problems with missing attachments in the SAP documents.
Workflow log creation after approval	Activate the **Turn on** parameter to create the workflow log when a document is approved.
Workflow log creation after recall	Activate the **Turn on** parameter to create the workflow log when a document is recalled.
Workflow log creation after reject	Activate the **Turn on** parameter to create the workflow log when a document is rejected.

Checks

/EBY/PDBO_EPC_CHECKS

Initial settings > Process parameters

Process	Description
Automatic workflow start after all checks succeed	Automatically starts the specified workflow on a document when the checks return no errors. In the parameters, enter the workflow ID(s) and activate **Turn on**.

Process	Description
Automatic workflow start after check failed	Automatically starts the specified workflow on a document when the checks return an error. In the parameters, enter the workflow ID(s) and activate **Turn on**.

Posting

/EBY/PDBO_EPC_POST

Initial settings > Process parameters

Process	Description
Add only the current workflow to the Smart Form	In the parameters, activate **Process current workflow only** to include only the details of the current workflow in the workflow log. If this option is not activated, details of all workflows to which the document was previously sent, as well as the current workflow, are listed in the workflow log.
Automatic post after workflow approval	Automatically posts the document when the specified workflow(s) is approved. In the parameters, enter the workflow ID(s) and activate **Turn on**.
Automatic workflow start after post failed	Automatically starts the specified workflow(s) on a document when posting of the document fails. In the parameters, enter a workflow ID and activate **Turn on**.
Automatic workflow start after post succeeded	Automatically starts the specified workflow(s) on a document when posting of the document is successful. In the parameters, enter a workflow ID and activate **Turn on**.

Process	Description
Smartform and document type for archiving after posting	Automatically adds the workflow log as an attachment to the PROCESS DIRECTOR document and the corresponding SAP document when the document is posted. In the parameters, enter the archiving document type and the name of the Smart Form that should be used to create the workflow log. Select a status and/or substatus to create the workflow log only when the document has this status/substatus.
Text ID of the email	
Wait x seconds for audit log archiving to finish	Enables you to delay archiving for a specified number of seconds to ensure that the PROCESS DIRECTOR workflow log and notes are archived correctly and available in the corresponding SAP documents. Enter a value here if you are experiencing problems with missing attachments in the SAP documents.

Workflow

`/EBY/PDBO_EPC_WORKFL`

Initial settings > Process parameters

Process	Description
Add only the current workflow to the Smart Form	In the parameters, activate **Process current workflow only** to include only the details of the current workflow in the workflow log. If this option is not activated, details of all workflows to which the document was previously sent, as well as the current workflow, are listed in the workflow log.

Process	Description
Archiving object parameters	Automatically creates a log of all activities that took place on a document during a workflow and adds this as an attachment to the PROCESS DIRECTOR document and the corresponding SAP document. In the parameters, enter the archiving document type and the name of the Smart Form that should be used to create the workflow log. Select a status and/or substatus to create the workflow log only when the document has this status/substatus. See also Workflow log creation.
Automatic post after workflow approval	Automatically posts the document when the specified workflow(s) is approved. In the parameters, enter the workflow ID(s) and activate **Turn on**.
Automatic workflow start after all checks succeed	Automatically starts the specified workflow on a document when the checks return no errors. In the parameters, enter the workflow ID(s) and activate **Turn on**.
Automatic workflow start after check failed	Automatically starts the specified workflow on a document when the checks return an error. In the parameters, enter the workflow ID(s) and activate **Turn on**.
Automatic workflow start after post failed	Automatically starts the specified workflow(s) on a document when posting of the document fails. In the parameters, enter a workflow ID and activate **Turn on**.
Automatic workflow start after post succeeded	Automatically starts the specified workflow(s) on a document when posting of the document is successful. In the parameters, enter a workflow ID and activate **Turn on**.

Process	Description
Automatic workflow start after transfer failed	Automatically starts the specified workflow(s) on a document when transfer of the document from the capture software to PROCESS DIRECTOR fails. In the parameters, enter a workflow ID and activate **Turn on**.
Automatic workflow start after transfer succeeded	Automatically starts the specified workflow(s) on a document when transfer of the document from the capture software to PROCESS DIRECTOR is successful. In the parameters, enter a workflow ID and activate **Turn on**.
Smart Form and document type for archiving after posting	Automatically adds the workflow log as an attachment to the PROCESS DIRECTOR document and the corresponding SAP document when the document is posted. In the parameters, enter the archiving document type and the name of the Smart Form that should be used to create the workflow log. Select a status and/or substatus to create the workflow log only when the document has this status/substatus.
Where to go after workflow approve? (advanced)	
Where to go after workflow recall? (advanced)	
Where to go after workflow reject? (advanced)	
Workflow log creation after approval	Activate the **Turn on** parameter to create the workflow log when a document is approved.
Workflow log creation after recall	Activate the **Turn on** parameter to create the workflow log when a document is recalled.

Process	Description
Workflow log creation after reject	Activate the **Turn on** parameter to create the workflow log when a document is rejected.

Other

/EBY/PDBO_EPC_OTHER

Initial settings > Process parameters

Process	Description
Document rejection: note/reason required	In the parameters, activate **Rejection note required** or/and **Rejection reason required** to ensures that users enter a predefined rejection reason, a note, or both when they reject a document. See also Configuring rejection reasons.
Parameters regarding uploaded files	These parameters are required to enable line item upload or document creation from a file.
Unlock original object?	

Mechanisms for populating fields

PROCESS DIRECTOR offers these mechanisms for populating business document fields with values:

Mechanism	When to use
Conversion functions	To populate display fields. Note that these values will *not* be persisted.

Mechanism	When to use
Presets	To automatically initialize document data. Presets are a key mechanism in processes driven by internal requests, such as managing Requisitions, However, they can also be applied during document-driven processes, for example when an external document is received. Use presets if you want certain document fields to be automatically populated with default values on document creation or update. The most important types are: ▪ fixed value presets ▪ dynamic presets (developed as customer coding) Each preset can only populate a single field. Apply them only to document model fields—for populating display fields, use conversion functions. Presets are always defined or coded as per individual customer requirement.
User exits / BAdIs	Customer implementations of the PROCESS DIRECTOR user exit / BAdI **Initialize values** can automatically initialize document data, as an alternative to dynamic presets. However, there are also other user exits / BAdIs that can manipulate the document model.
Determinations	To automatically complete document data. Determinations are a key mechanism in processes driven by incoming documents. Use determinations if you want the values of certain document fields to be calculated during a human task, for example, when checking or updating a document. A determination can populate a single field or multiple fields, depending on the determination. Although it is possible to develop customer determinations, usually you can accommodate the customer's business scenario by configuring the determinations already available out-of-the-box for the given PROCESS DIRECTOR process type.

Mechanism	When to use
Entry templates	To manually enter document data. Use entry templates if you want to let users define their own sets of default values that they can insert during manual processing to speed up data entry. You can also define system templates which will be available to all users. The most important types are: ▪ line item templates ▪ account templates An entry template usually populates several fields. Entry templates are always defined as per individual customer requirement.
External data	This is the main source of field input for document-driven process types.
Customer events	Although customer events are not supported in the standard configuration mode, they are mentioned here for the sake of completeness. Among other capabilities, customer events can manipulate a business document's data in arbitrary ways.

Workflow control patterns and step types

Control patterns

Parallel split (AND-split)

A parallel split may occur when several workflow steps are configured with the same previous step, that is, the previous step has several successors, or when the workflow has several start steps. Valid successors proceed in parallel and may or may not be synchronized at some future time. A parallel split may cause the workflow to have more than one final step.

Synchronization (AND-join)

A synchronization may occur when a workflow step is configured with several previous steps. All valid previous steps must be approved for the successor to proceed.

Step types

Valid step

A step is valid by default, but can be invalidated at runtime in case any specified step conditions are not satisfied.

Note: The validity status of a step can be overridden by an implementation of the user exit / BAdI **Workflow steps handling** interface method `CHECK_STEP_VALIDITY`.

Start step

A valid step not configured with a previous step is a start step. In other words, a successor step can never become a start step by way of an invalid previous step. If several start steps exist, a parallel split occurs.

If *all* start steps of a workflow have the processor assignment `In background (automatic, without manual interaction)`, it is possible for an automatic workflow start to occur.

If it is possible for a step to be a start step, you *must* ensure that the step's **Reject** setting is set to `1 Cancel the workflow`.

If a workflow doesn't have at least one start step, it cannot be started.

Final step

A valid step is final if it is not defined as a previous step or if none of its successors are valid.

For a workflow to be approved, *all* of its final steps must be approved.

Standard documentation objects

The following documentation objects are available in the PROCESS DIRECTOR standard for use in messages and workflow emails. Most of these documentation objects include placeholders, which are replaced by the relevant values when the message or email that contains the documentation object is sent. Documentation objects can be created or edited in transaction `SE61`.

Use	Documentation object name	Purpose
Messages	`/EBY/PDBO_DISPATCH_CUSTOMER`	Insert customer address data
	`/EBY/PDBO_DISPATCH_SAMPLE`	Sample data for messages
	`/EBY/PDBO_DISPATCH_USER`	Insert user address data
Posting notification	`/EBY/PDBO_DISPATCH_POST`	Inform the creator of a document that the document has been posted.

Use	Documentation object name	Purpose
	/EBY/PDMDVM_DISPATCH_POST	Inform the creator of a vendor master data maintenance document that the document has been posted.
Collective emails	/EBY/PDWC_COLLECTIVEREMINDER	Text for collective reminder mails. A collective email is sent when the program /EBY/PDWC_DUE_DATE_CHECK is run. The collective email contains all workflow step notifications for that processor since the last program run.
	/EBY/PDWC_COLLECTIVE_HTML	Same as /EBY/PDWC_COLLECTIVEREMINDER, but in HTML format
Workflow reminder emails	Use objects beginning with /EBY/ICWC for Accounts Payable documents and objects beginning with /EBY/PDWC for all other document types.	
	/EBY/PDWC_REMINDER /EBY/ICWC_REMINDER	Remind a workflow processor that a workflow step is overdue.
	/EBY/PDWC_REMINDER_HTML /EBY/ICWC_REMINDER_HTML	Same as/EBY/PDWC_REMINDER / /EBY/ICWC_REMINDER, but in HTML format
	/EBY/PDWC_SHORT_REMINDER /EBY/ICWC_SHORT_REMINDER	Short text included in a collective mail to remind a workflow processor that a workflow step is overdue.
	/EBY/PDWC_SHORT_REMINDER_HTM /EBY/ICWC_SHORT_REMINDER_HTM	Same as /EBY/PDWC_SHORT_REMINDER / /EBY/PDWC_SHORT_REMINDER, but in HTML format
Workflow receipt emails	Use objects beginning with /EBY/ICWC for Accounts Payable documents and objects beginning with /EBY/PDWC for all other document types.	
	/EBY/PDWC_SENT /EBY/ICWC_SENT	Inform a workflow processor that they have received a document for processing.

Use	Documentation object name	Purpose
	/EBY/PDWC_SENT_HTML /EBY/ICWC_SENT_HTML	Same as /EBY/PDWC_SENT / /EBY/ICWC_SENT, but in HTML format
	/EBY/PDWC_SHORT_SENT_HTML /EBY/ICWC_SHORT_SENT_HTML	Short text included in a collective email to inform the user that a document has been sent to a workflow
Workflow query emails *	/EBY/PDWC_QUERY	Inform the recipient of a query that a query has been sent to them.
	/EBY/PCWC_QUERY_HTML	Same as /EBY/PDWC_QUERY, but in HTML format.
Workflow approval emails *	/EBY/PDWC_WF_APPROVED	Inform the workflow initiator that a workflow has been approved.
	/EBY/PDWC_WF_APPROVED_HTML	Same as /EBY/PDWC_WF_APPROVED, but in HTML format.
	/EBY/PDWC_SHORT_WF_APPROVED	Short text included in a collective mail to inform the workflow initiator that a workflow has been approved.
	/EBY/PDWC_SHORT_WF_APPR_HTML	Same as /EBY/PDWC_SHORT_WF_APPROVED, but in HTML format.
Workflow recall emails *	/EBY/PDWC_WF_RECALLED	Inform the workflow processors that a workflow has been recalled.
	/EBY/PDWC_WF_RECALLED_HTML	Same as /EBY/PDWC_WF_RECALLED, but in HTML format.
	/EBY/PDWC_SHORT_WF_RECALLED	Short text included in a collective mail to inform the workflow processors that a workflow has been recalled.

Use	Documentation object name	Purpose
	/EBY/PDWC_SHORT_WF_RECL_HTML	Same as /EBY/PDWC_SHORT_WF_RECALLED, but in HTML format.
Workflow cancelation emails *	/EBY/PDWC_WF_CANCELLED	Inform the workflow initiator that a workflow has been canceled as the result of a step being rejected (**Reject** setting is set to **Cancel the workflow** in the step settings).
	/EBY/PDWC_WF_CANCELLED_HTML	Same as /EBY/PDWC_CANCELLED, but in HTML format
	/EBY/PDWC_SHORT_WF_CANCELLED	Short text included in a collective email to inform the workflow initiator that a workflow has been canceled as the result of a step being rejected.
	/EBY/PDWC_SHORT_WF_CANC_HTML	Same as /EBY/PDWC_SHORT_WF_CANCELLED, but in HTML format.
MOBILE APPROVAL emails*	/EBY/PDWC_SENT_MA	Inform a MOBILE APPROVAL workflow processor that they have received a document for processing.
	/EBY/PDWC_SENT_HTML_MA	Same as /EBY/PDWC_SENT_MA, but in HTML format
	/EBY/PDWC_QUERY_MA	Inform a MOBILE APPROVAL user that a query has been sent to them.
	/EBY/PDWC_QUERY_HTML_MA	Same as /EBY/PDWC_QUERY_MA, but in HTML format
	/EBY/MOBILE_APPR_OK	Inform a MOBILE APPROVAL workflow processor that their action (approve, reject, add note) was successful.

Use	Documentation object name	Purpose
	/EBY/MOBILE_APPR_ERR	Inform the workflow processor that their action (approve, reject, add note) could not be performed.
Other	/EBY/PDFI_GLPR_REMINDER	Inform the creator of an FI provision document that the validity date of the posted document has been exceeded.

* Not available for Accounts Payable documents.

Placeholders for documentation objects

Placeholders can be used to insert variable data. The placeholder is replaced with the relevant value. For example, in a message text, the placeholder `&SENDER-ADDRESS-FIRSTNAME&` will be replaced in the output by the first name of the user that creates the message and the placeholder `&SENDER-ADDRESS-LASTNAME&` will be replaced by the last name of the user.

These placeholders are available for use in workflow and message texts:

Use	Placeholder	Inserts in the email
User data	&SENDER-ADDRESS-[FIELD]&	You can use any field provided in the SAP structure BAPIADDR3 with this placeholder. Replace [FIELD] with the name of the field. **Examples**: &SENDER-ADDRESS-FIRSTNAME& &SENDER-ADDRESS-LASTNAME& &SENDER-ADDRESS-E_MAIL& &SENDER-ADDRESS-TEL1_NUMBR& &SENDER-ADDRESS-TEL1_EXT& &SENDER-ADDRESS-FAX_NUMBR& **Note**: For workflow texts, the sender will be the current step's processor or initiator.

Use	Placeholder	Inserts in the email

	`&RECIPIENT-ADDRESS-[FIELD]&`	You can use any field provided in the SAP structure `BAPIADDR3` with this placeholder. Replace `[FIELD]` with the name of the field.
		Examples:
		`&RECIPIENT-ADDRESS-TITLE_P&`
		`&RECIPIENT-ADDRESS-FIRSTNAME&`
		`&RECIPIENT-ADDRESS-LASTNAME&`
		`&RECIPIENT-ADDRESS-DEPARTMENT&`

Document data	`&HEADER-[FIELD]&`	You can use any field provided in the header interface structure of the document type with this placeholder. For example, for Requisitions, use any field in the structure `/EBY/PDPO_SHDR_IF`, for Customer Orders the structure `/EBY/PDSO_SHDR_IF`, etc. Replace `[FIELD]` with the name of the field. **Examples**: `&HEADER-OBJ-TXT&` (document description) `&HEADER-NUMBR&` (document number) `&HEADER-GUID&` (unique document ID) See the Process types reference (*Document model customization*) for more information on process type specific structures.
Workflow data	`&WORKFLOW-WC_DESCR&`	Description of the current workflow.
	`&WORKFLOWSTEP-WC_STEP_DESCR&`	Description of the current workflow step.
	`&WORKFLOWSTEP-WC_STEP_DUEDATE&`	Due date of the current workflow step.

Use	Placeholder	Inserts in the email
	&RECEIPT_LIST&	List of workflows received for processing.
	&REMINDER_LIST&	List of overdue workflows.
	&APPROVED_WORKFLOW_LIST&	List of approved workflows.
	&APPROVED_WORKFLOW_LIST-COUNTER&	Number of approved workflows.
	&RECALLED_WORKFLOW_LIST&	List of recalled workflows.
	&RECALLED_WORKFLOW_LIST-COUNTER&	Number of recalled workflows.
	&CANCELLED_WORKFLOW_LIST&	List of workflows canceled after rejection of a step.
	&CANCELLED_WORKFLOW_LIST-COUNTER&	Number of workflows canceled after rejection of a step.
	&LAST_NOTE&	Text of the last note that was added to the document by the current workflow processor.
MOBILE APPROVAL data	&MOBILE-APPROVE&	Link that the MOBILE APPROVAL user clicks to approve the document.
	&MOBILE-APPROVE_HTML&	Same as &MOBILE-APPROVE&, but in HTML format. May be displayed as a button instead of a link.
	&MOBILE-REJECT&	Link that the MOBILE APPROVAL user clicks to reject the document.
	&MOBILE-REJECT_HTML&	Same as &MOBILE-REJECT&, but in HTML format. May be displayed as a button instead of a link.

Use	Placeholder	Inserts in the email
	`&MOBILE-NOTE&`	Link that the MOBILE APPROVAL user clicks to add a note to the document.
	`&MOBILE-NOTE_HTML&`	Same as `&MOBILE-NOTE&`, but in HTML format. May be displayed as a button instead of a link.
Other	`&URL&`	The Web Application URL specified in Workflow General Settings.
	`&HTML_EMAIL_CSS_STYLES&`	Add this alias to the <body> section of your own HTML email documentation objects to use your own stylesheet for HTML emails. See Customizing HTML emails for more information. **Note**: This placeholder applies only to HTML emails.

To include a single document link in a text, use this syntax:

- For Accounts Payable documents: `&URL&&&GUID=&HEADER-WEBGUID&`
- For all other document types: `&URL&&ACTION=OTHER~&GUID=&HEADER-GUID&`

To change the default Worklist node in which the document will appear when the single document link is followed, you can create a customer implementation of the user exit / BAdI **Set Worklist node of a single document link**.

User profile parameters

You can add the following PROCESS DIRECTOR parameters to an SAP user profile:

Parameter	Description
`/EBY/PDBO_DEBUG_MSG`	Adds debugging messages to the system messages. See the *PROCESS DIRECTOR Troubleshooting Guide* for more information.

Parameter	Description
/EBY/PDBO_CALLSTACK	Displays message call stacks. See the *PROCESS DIRECTOR Troubleshooting Guide* for more information.
/EBY/PDBO_PROJECT	Assigns a user to a project. The project configuration will be used instead of the standard configuration for this user. See the *PROCESS DIRECTOR SAP Configuration Guide* for information on working with projects.
/EBY/PDBO_CHK_SPLITV	Set to a value of X, this parameter splits the checks and determinations configuration screen vertically so that the check details and message parameters are displayed on the right instead of at the bottom of the screen. This is convenient when using a high resolution screen.

Programs

The following programs are available for all PROCESS DIRECTOR process types. Use the standard SAP transaction SE38 to run a program once, or SM36 to schedule it as a background job.

- /EBY/PDWC_DUE_DATE_CHECK

  Sends receipt emails when users receive documents in a workflow and sends reminder emails for overdue workflows. Emails can also be sent when workflows are approved, recalled, or canceled after rejection.

- /EBY/PDBO_REPETITOR

  Automatically checks and posts PROCESS DIRECTOR documents.

- /EBY/PDBO_CONFIG_DUMP

  Use this program to download and upload PROCESS DIRECTOR project configurations to XML files and compare configurations.

- /EBY/PDBO_CONFIG_GEN_WORKLIST

  Generates the Worklist.

- /EBY/PDBO_CHECK_NESTEDSET_NEW

  Regenerates the mapping of documents to static Worklist nodes.

- `/EBY/PDBO_GEN_FAILED_NSTDSETS`

  Generates the mapping of skipped documents to static Worklist nodes.

- `/EBY/PDBO_CHECK_WL_DOC_COUNT`

  Checks or restores Worklist document counters for static nodes.

- `/EBY/PDBO_CONFIG_WRKL_ENV_CONV`

  Sets the Worklist node **Not visible in** option for nodes created prior to PROCESS DIRECTOR 7.3.

- `/EBY/PDBO_CHANGE_ARC_LINKS`

  Changes the archiving link of attachments to a different process type.

See the following sections for information on process type specific programs:

Programs for Accounts Payable

Programs for Financial Postings

Programs for Payment Advices

Programs for Payment Approvals

/EBY/PDWC_DUE_DATE_CHECK

Use the `/EBY/PDWC_DUE_DATE_CHECK` program to send receipt emails when users receive documents in a workflow and to send reminder emails for overdue workflows. Emails can also be sent when workflows are approved, recalled, or canceled after rejection.

Note: Customers using ReadSoft's WORK CYCLE for processing Accounts Payable documents can assign email users to workflow steps. Currently, PROCESS DIRECTOR does not support email users, so these types of users are not processed by the `/EBY/PDWC_DUE_DATE_CHECK` program and will not receive email notifications.

Setting	Description
Object type	Process type for which emails will be sent. Select the blank entry at the end of the list to include all process types.
PD document number	Range of PD documents to be processed.

Setting	Description
Approved workflows	Select to send emails to the workflow initiator when a workflow is approved.
Rejected workflows	Select to send emails to the workflow initiator when a workflow step is rejected, with the result that the entire workflow is canceled (**Reject** setting is set to **Cancel the workflow** in the step settings).
Recalled workflows	Select to send emails to workflow processors when a workflow is recalled.
Receipt emails	Select to send emails when users receive a document for processing in a workflow.
Reminder emails	Select to send emails when a workflow step is overdue.
Step overdue from	Enter a date. Emails will be sent for workflow steps that become overdue on this date or between this date and the date the program is run.
Reminder interval	Interval, in days, at which further reminder mails should be sent. For example, if you enter 3, reminder mails are sent every three days.
Ignore next reminder	Select if the next due reminder email should not be sent. For example, if the next reminder email is due on Friday and you run the `/EBY/PDWC_DUE_DATE_CHECK` program on Thursday, no reminder will be sent on the Friday. However, a reminder email is sent directly after the program has run.
Collective email	One email containing all workflow step notifications for the processor since the last program run is sent.
Email per document	An individual email for each document is sent. **Note**: This setting only applies to receipt and reminder emails.

/EBY/PDBO_REPETITOR

Use the `/EBY/PDBO_REPETITOR` program to automatically check and post documents at regular intervals. The document status is automatically changed after checking/posting takes place. The program can also be called with the transaction `/EBY/PDBO_REPETITOR`.

General selection criteria

Setting	Description
Object type	Process type for which documents will be checked/posted.
PD document number	Document or range of documents to be checked/posted.
Document status	Document statuses to be checked. For example, it makes no sense to include documents that have already been posted.
Substate	Document substatuses to be checked.
Workflow	Workflow statuses of documents to be checked. For example, you may want to check only approved documents.
Created on	Date on which the document was created.
Changed on	Date on which the document was last changed.

Actions

Setting	Description
Check and post	Select to check the documents and post them if there are no errors.
Check only	Select to check the documents only.

Messages and application log

Setting	Description
Display messages	Select to display the messages generated during the program run.

Setting	Description
Create application log	Select to create an application log that you can view in transaction SLG1. Enter the object /EBY/PD and the subobject /EBY/PD_REPETITOR in SLG1.

/EBY/PDBO_CONFIG_DUMP

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
CUS&EPD_WRKLST	Exclude the Worklist entries from download
CUS&IPD_WRKLST	Download only the Worklist entries
CUS&I_COCKPIT	Download only PROCESS DIRECTOR Accounts Payable tables

Download XML file

Setting	Description
To file	Name of the XML file in which the configuration will be stored.

Setting	Description
Of document type	The document type to be downloaded. It includes independent tables of all the document types. When a table is document dependent: It contains all the entries where `OBJ` is empty. It contains all the entries where `OBJ` is equal to that specified on the selection screen.
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 **Compare** 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.

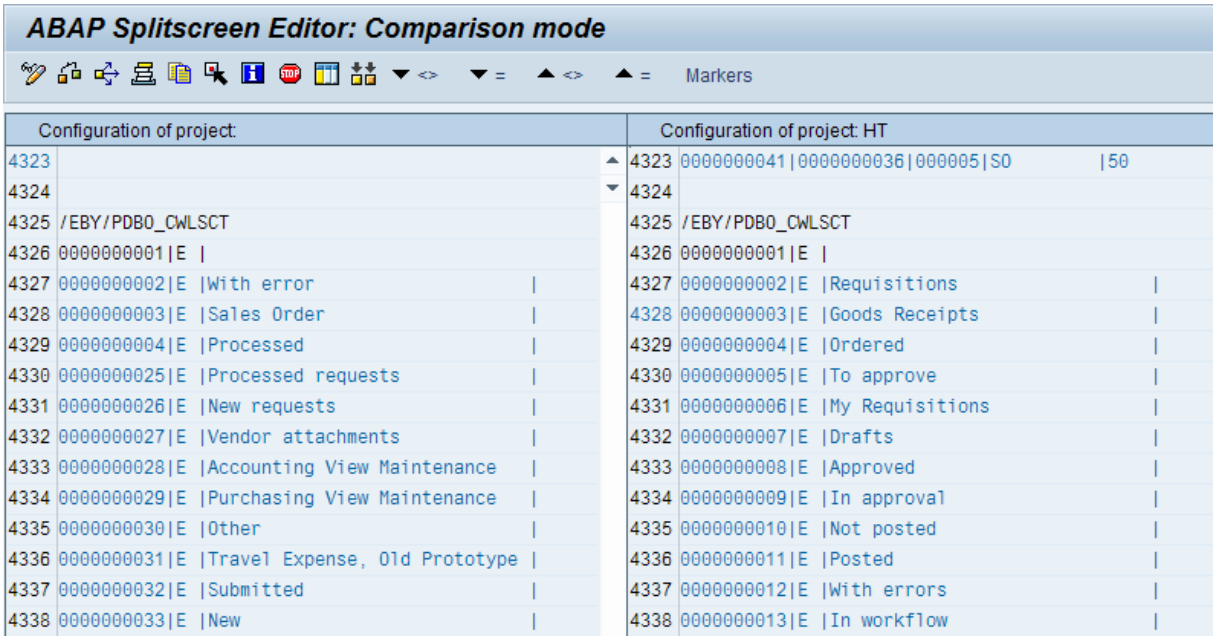
Setting	Description
Request/Task	Name of a customizing request. All uploaded data is included in this request, which can then be released and transported to another system.
From file	Name of the XML file from which the configuration will be uploaded.
To project	Name of the project to which the configuration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

Setting	Description
Remove before upload	Activate this option to remove existing entries in any PD or PDAP customizing table stored in the uploaded file. 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. If you deactivate this option: ▪ Entries that are present in the system database, but not in the XML file, are retained. ▪ Entries that are present in the XML file, but not in the system, are added to the database. ▪ Entries in the XML file overwrite identical entries in the database.

Compare

Setting	Description
Ignore field 'project'	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.
Include system tables	Activate this option to include system tables as well as customized tables in the comparison. If you are comparing XML files, the system tables must have been included when the configuration was downloaded to the XML file (see Include system tables).
Element 1, Element 2	Select the elements that you want to compare. For example, to compare a stored XML file with a project configuration in the system: In the **Element 1** section, select **Local file** and enter the file path and name. In the **Element 2** section, select **Customizing on server** and select a project.

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.



/EBY/PDBO_CONFIG_GEN_WORKLIST

Use the `/EBY/PDBO_CONFIG_GEN_WORKLIST` program to generate the PROCESS DIRECTOR Worklist. This can also be done manually in the Worklist configuration screen.

Parameter	Description
Document batch size	The number of documents that will be processed in one batch.
Remove orphan entries	Removes orphan entries from the table that contains the document assignment.

/EBY/PDBO_CHECK_NESTEDSET_NEW

Use the `/EBY/PDBO_CHECK_NESTEDSET_NEW` program to regenerate the mapping of documents to static Worklist nodes.

This program maps documents to the correct Worklist nodes, according to the nodes' ranges configuration. For example, if a node's ranges configuration specifies `STATE = 'OK'` (document status = posted), all documents with this status will be mapped to and therefore displayed in the node.

Normally, you only need to run this program if you are having problems with Worklist generation.

/EBY/PDBO_GEN_FAILED_NSTDSETS

Use the `/EBY/PDBO_GEN_FAILED_NSTDSETS` program to generate the mapping of skipped documents to static Worklist nodes.

When the Worklist is generated, documents that are currently locked because they are being edited by users are skipped by the Worklist generation program, which means that they may be displayed in the wrong node, or may not be displayed at all. These documents are written to the `/EBY/PDBO_TWLF` table. The `/EBY/PDBO_GEN_FAILED_NSTDSETS` program reads this table and generates the mapping to the correct Worklist nodes for these documents, provided the documents are no longer locked.

Normally, you only need to run this program if the Worklist generation messages indicate that documents were skipped. The program should be run repeatedly until all skipped documents are processed successfully.

/EBY/PDBO_CHECK_WL_DOC_COUNT

Use the `/EBY/PDBO_CHECK_WL_DOC_COUNT` program to check or restore Worklist document counters for static nodes.

This program calculates the current number of documents assigned to static nodes, checks these numbers against the database and displays any discrepancy errors. Check the **Repair** box before running the program if the new values should replace the old ones in the database.

Normally, you only need to run this program if you suspect that Worklist node counters are incorrect.

/EBY/PDBO_CONFIG_WRKL_ENV_CONV

In PROCESS DIRECTOR 7.3, by default the Worklist node **Not visible in** option is activated for all environments except SAP GUI and Web Application (such as the **Mobile Application** environment), so that the Worklist node is not displayed for those environments. If a node has been created prior to 7.3, this option is not activated by default and must be set retrospectively for all nodes by running the `/EBY/PDBO_CONFIG_WRKL_ENV_CONV` program. The option can then be deactivated individually for nodes that should be visible in that environment.

/EBY/PDBO_CHANGE_ARC_LINKS

Use the `/EBY/PDBO_CHANGE_ARC_LINKS` program to assign archiving links for attachments to a different PROCESS DIRECTOR object type.

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 Advanced Configuration Guide* for more information on developing BAdI implementations.

BAdI	Description
Basic	
Screen PBO/PAI	Add custom logic for the screen display. For example, you can use this BAdI to display or verify values.
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	
Set messages	Add custom logic to modify messages sent to the Web Application.
Set actions	Add custom logic to modify which actions are available to the Web Application.
User logon	Add custom logic to modify the behavior of the application during logon to the Web Application.

BAdI	Description
Workflow	
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.

BAdI Screen PBO/PAI

User exits / BAdIs > Platform > Basic

This platform user exit / BAdI currently provides the interface methods `PBO`, `PAI`, `SET_PAI_OKCODE` and `EXCLUDE_FUNCTION_CODES` that you can implement to add your own customer logic for the screen display. For example, you can use this BAdI to display or verify values.

PBO method

Calling sequence

This method is called in the SAP GUI when the screen is refreshed after an action is processed.

Method signature

Type	Parameter	Description
	`IR_SELOBJECT`	Selected object
	`IR_MAINSCREEN`	Main screen object
	`IC_EDITMODE`	GUI edit mode
	`ERROR_OCCURRED`	

PAI method

Calling sequence

This method is called in the SAP GUI before an action is processed.

Method signature

Type	Parameter	Description
	IR_SELOBJECT	Selected object
	IR_MAINSCREEN	Main screen object
	CC_OKCODE	Screens, function code triggered by PAI
	ERROR_OCCURRED	

SET_PAI_OKCODE method

Calling sequence

This action is called in the SAP GUI before an action is processed.

Method signature

Type	Parameter	Description
	IR_SELOBJECT	Selected object
	IR_MAINSCREEN	Main screen object
	CC_OKCODE	Screens, function code triggered by PAI
	ERROR_OCCURRED	

EXCLUDE_FUNCTION_CODES method

Calling sequence

This method is called in the SAP GUI when the screen is refreshed after an action is processed.

Method signature

Type	Parameter	Remarks
	IT_UCOMM_MAPPING	Map GUI commands to actions
	IT_SELECTED_OBJECTS	Table of /EBY/CL_PDBO (any PD document)
	IR_MAINSCREEN	Main screen object
	CT_EXCLUDED_FCODES	This parameter is pre-populated with the list of function codes excluded for any of these reasons: ▪ not assigned to an action ▪ it is the function code of a button that calls the detail screen, but we are already in the detail screen ▪ because of state ▪ not authorized It is possible to re-enable a disabled action by deleting its function code from CT_EXCLUDED_FCODES. To do this, you need to know the function code mapping of the action.
	ERROR_OCCURRED	

BAdI Modify Worklist

User exits / BAdIs > Platform > Basic

This platform user exit / BAdI currently provides the interface method MODIFY_WORKLIST that you can implement to add your own customer logic for the PROCESS DIRECTOR Worklist configuration.

Typical usages of this user exit / BAdI are to:

- change the currently selected Worklist node

- change the Worklist construction, for example, by disabling some nodes (greying them out) or removing them entirely

MODIFY_WORKLIST method

Calling sequence

In the SAP GUI environment, this user exit is called during PROCESS DIRECTOR startup and when the **Refresh** toolbar button is clicked (action `REFRESH_ALL_OBJECTS Refresh list of objects`).

1. At PROCESS DIRECTOR startup when the Worklist tree is built.
2. At PROCESS DIRECTOR startup when the user's previous Worklist node is restored (`action=STARTPROCESSDIRECTOR`, `/EBY/CL_PDVI_FORW_REFRESH` does `RAISE EVENT refresh_data` which triggers `/EBY/CL_PDVI_WORKLIST_TREE->EXPEND_NODES_FOR_SELECTED` which calls `REFRESH_TREE_FULL`).
3. After pressing refresh button (`fcode=REFRESH`, `action=REFRESH_TREE_AND_OBJ`, `/EBY/CL_PDVI_FORW_RSH_WRK_TREE` does `RAISE EVENT refresh_tree` which triggers `/EBY/CL_PDVI_MAINSCREEN->ON_REFRESH_TREE` which calls `REFRESH_TREE_FULL`).

`REFRESH_TREE_FULL` calls the BAdI.

In the Web Application environment, this user exit is called from the action `WA_GET_TREE_DATA Retrieve Worklist data XML`, immediately after retrieving the last Worklist node visited by the user.

Method signature

Type	Parameter	Remarks
	`IR_CONTEXT`	Reference to the user session context
	`IR_WORKLIST`	
	`CR_SELECTED_NODE`	
	`ERROR_OCCURRED`	

Changing the currently selected Worklist node

To change the currently selected node, simply assign a Worklist node to the `CHANGING` parameter `CR_SELECTED_NODE`.

Note: Under certain conditions, such reassignments of the currently selected Worklist node can be overridden by an implementation of the user exit / BAdI **Set Worklist node of a single document link**.

One way to obtain an existing Worklist node is to search for it by node ID via the method `IR_WORKLIST->FIND_NODE2`. This only works for static nodes and semi-dynamic nodes which are not under dynamic nodes, because this method uses the static node ID taken from the Worklist configuration.

Changing the Worklist construction

Removing nodes completely is not recommended, since you can lose access to a document in case a change in its properties causes it to be assigned to a removed node.

Warning: Document access can be lost even if a removed node becomes visible again after the BAdI implementation is deactivated, because the document won't be visible in that node until the next update action is performed on it—which can never happen unless document is visible in some other node.

Note: Since this user exit / BAdI is called *after* Worklist node authorization checks have been performed, any nodes added to the Worklist by a **Modify Worklist** implementation are *not* subject to authorization checks (unless you explicitly perform authorization checks in your coding).

To make a Worklist node invisible, Call `IR_WORKLIST->GET_NODE_FOR_EXT_ID` to find a node using its external ID, and then set `MB_INVISIBLE = ABAP_TRUE` on that node.

BAdI Modify field profiles

User exits / BAdIs > Platform > Basic > Modify field profiles

This user exit / BAdI currently provides the interface method `MODIFY_FIELD_STATES` that you can implement to add your own customer logic when field statuses are applied.

MODIFY_FIELD_STATES method

Method signature

Type	Parameter	Remarks
	`IC_OBJ`	Object type
	`IC_COMP_TYPE`	Screen component type
	`IC_GRID_NO`	Grid number

Type	Parameter	Remarks
	IC_DMILEVEL	Logical level of the internal data model
	IC_GRID_CONF	Grid configuration
	IT_OBJECTS	Table of objects
	IR_CONTEXT	Reference to the user session context
	CR_STATES	Field states / field profile

BAdI Set Worklist node of a single document link

User exits / BAdIs > Platform > Basic

When a user clicks a single document link in a workflow notification email or SAP Business Workflow work item, the document will open up in a default Worklist node that is calculated from the document GUID.

This platform user exit / BAdI currently provides the interface method SET_NODE that you can implement to add your own calculation routine for determining this default Worklist node.

You can change to any node that the document belongs to.

Note: The document will not be displayed in the overview list if you change to a node that does not contain the document as part of its document selection.

SET_NODE method

Calling sequence

This method is only called when resolving SAP Business Workplace work items and single document links in workflow notification emails. This method is called *after* the user exit / BAdI **Modify Worklist**.

Method signature

Type	Parameter	Remarks
	IR_CONTEXT	Reference to the user session context

Type	Parameter	Remarks
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method.
	CR_NODE	Worklist node

BAdI Determination results

User exits / BAdIs > Platform > Basic > Determination results

This user exit / BAdI currently provides the interface method PROCESS that you can implement to add your own customer logic after determinations have been performed.

PROCESS method

Calling sequence

This method is called after determinations have been performed, but before checks run.

Method signature

Type	Parameter	Remarks
	IC_OBJ_TYPE	Document type
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IR_CONTEXT	Action chain context
	IS_BO_HEADER	Basic document header
	CR_OBJECT	Document object

BAdI Rejection reason

User exits / BAdIs > Platform > Basic > Rejection reason

This user exit / BAdI currently provides the interface methods BEFORE_REJECT and AFTER_REJECT that you can implement to add your own customer logic before and after a rejection reason is entered.

BEFORE_REJECT method

This method can be used, for example, to exclude mandatory rejection reasons/notes for a specific company code, or change the rejection reason descriptions.

Calling sequence

This method is called before the **Rejection reason** popup is displayed.

Method signature

Type	Parameter	Remarks
	`IR_OBJECT`	Abstract PROCESS DIRECTOR basis object
	`CS_REJECT_REQ`	Document rejection requirements
	`CT_REASON`	Document rejection reason descriptions

AFTER_REJECT method

This method can be used, for example, to change values in the document or create a PDF note after document rejection.

Calling sequence

This method is called after the user selects a rejection reason.

Method signature

Type	Parameter	Remarks
	`IR_OBJECT`	Abstract PROCESS DIRECTOR basis object

BAdI Set messages

User exits / BAdIs > Platform > Web Application

This user exit / BAdI currently provides the interface method `SET_MESSAGES` that you can implement to add your own customer logic to modify messages sent to the Web Application.

SET_MESSAGES method

Method signature

Type	Parameter	Description
	IR_CONTEXT	Reference to the user session context
	IR_WA_CONTEXT	Web Application session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method.
	CT_MESSAGES	Messages with localized text

BAdI Set actions

User exits / BAdIs > Platform > Web Application

This user exit / BAdI currently provides the interface method SET_ACTIONS that you can implement to add your own customer logic to modify which actions are available to the Web Application. You can use this method to enable any action, even if it is not in the Web Application's base action pool.

SET_ACTIONS method

Method signature

Type	Parameter	Description
	IR_CONTEXT	Reference to the user session context
	IR_WA_CONTEXT	Web Application session context
	CT_ACTIONS	List of actions for the Web Application
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method.

BAdI User logon

User exits / BAdIs > Platform > Web Application

This user exit / BAdI currently provides the interface methods `SSO_USER_CHECK`, `USER_CHECK` and `USER_PW_CHANGE` that you can implement to add your own customer logic to modify the behavior of the application during logon to the Web Application.

PROCESS DIRECTOR provides a standard implemention for Accounts Payable, `/EBY/ICWC_USER_LOGON`, which should be activated if PROCESS DIRECTOR should call User Exit 923, `User check during logon to the Web Application`, when users log on to the Web Application or change their password (either the initial password or the password in the user profile).

SSO_USER_CHECK method

Calling sequence

This method is called during user logon to the Web Application from a mobile device. It can be used to extend the search logic for the user ID, and indicates whether or not a user can access the Web Application from a mobile device.

Method signature

Type	Parameter	Description
	`VALUE( IC_SSO_ID )`	SSO ID used for logon to the Web Application
	`VALUE( IC_USER_ID )`	ID used for logon to the Web Application
	`VALUE( IC_USER_TYPE )`	PROCESS DIRECTOR user type
	`VALUE( IT_TMAL )`	Table of logons with SSO ID
	`CB_ALLOWED`	

USER_CHECK method

Calling sequence

This method is called during user logon to the Web Application. It can be used to provide additional logic for authenticating the user, or to replace the user who logs in with an

alternative user. In the latter case, checks such as checking the password, or checking whether the account is locked, are performed with the logon user ID, but authorization checks are performed against the data of the alternative user. For example: Dave has authorization to display the Financial Postings Worklist node, but Roger does not. Dave logs on to the Web Application, but the Financial Postings node is not displayed because Roger (the alternative user) does not have authorization to display it.

Method signature

Type	Parameter	Description
	`VALUE( IC_USER_ID )`	Current processor
	`VALUE( IC_USER_TYPE )`	PROCESS DIRECTOR user type
	`VALUE( IC_PASSWORD )`	PROCESS DIRECTOR user password type
	`CS_USER_DATA`	Structure containing user data
	`CS_MESSAGE`	Return parameter
	`CB_NO_STANDARD`	

USER_PW_CHANGE method

Calling sequence

This method is called when users change their password in the Web Application. It can be used to extend the logic for authenticating the user.

Method signature

Type	Parameter	Description
	`VALUE( IC_USER_ID )`	Current processor
	`VALUE( IC_USER_TYPE )`	PROCESS DIRECTOR user type

Type	Parameter	Description
	VALUE(IC_PASSWORD)	PROCESS DIRECTOR user password type
	VALUE(IC_NEW_PASSWORD)	PROCESS DIRECTOR user password type for new password
	CS_MESSAGE	Return parameter
	CB_NO_STANDARD	

BAdI Workflow email handling

User exits / BAdIs > Platform > Workflow

This user exit / BAdI currently provides the interface method BEFORE_CREATE_MAIL that you can implement to add your own customer logic for handling workflow emails.

BEFORE_CREATE_MAIL method

Calling sequence

This method is called before a workflow email is created. Workflow emails are sent out by the /EBY/PDWC_DUE_DATE_CHECK program.

Method signature

Type	Parameter	Description
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration

Type	Parameter	Description
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data
	IC_STEP_ID	Workflow step ID. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_CONF	Structure containing workflow step configuration. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_DATA	Structure containing workflow step data. This parameter refers to the step that is being processed by the BAdI method, by the user.
	IN_STEP_POSITION	This value represents the position of the step that is currently being processed by the user. **Note**: When the workflow is in the start phase but has not yet started, the value is set to 0. The same value is set when the workflow has already finished.
	IT_STEP_USER_TYPES	User types available for the step. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IT_STEP_PROCESSORS	List containing all processors of the current and next steps
	IT_STEP_APPROVERS	List containing all step approvers of the current and previous steps
	IT_NEXT_STEPS	List containing all next steps with their details
	IT_CURR_STEPS	List containing all current steps with their details
	IT_PREV_STEPS	List containing all previous steps with their details

Type	Parameter	Description
	IR_OBJECT	Base class of all ABAP objects
	CB_CANCEL_MAIL	Indicates if the email should be created
	CS_SENDER	PROCESS DIRECTOR user and user type
	CT_RECIPIENTS	List of user IDs of email recipients
	CT_CC_RECIPIENTS	List of user IDs of email CC recipients
	CC_SUBJECT	Email subject
	CT_TEXT	Email text
	CC_LANGU	Email language
	CT_ALIASES	Aliases and their replacements
	CC_MAIL_PURPOSE	Email purpose
	CC_EMAIL_TYPE	Email type

BAdI Workflow handling

User exits / BAdIs > Platform > Workflow

This user exit / BAdI currently provides the interface methods BEFORE_START, AFTER_START and AFTER_FINISH that you can implement to add your own customer logic for handling workflows.

This user exit / BAdI is filter-dependent and allows you to filter on object (process type) and workflow.

BEFORE_START method

This method can be used, for example, to determine whether the workflow is valid or not.

Calling sequence

This method is called before a workflow starts.

Method signature

Type	Parameter	Description
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data
	IR_WORKFLOW	Reference to the workflow object
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type and workflow ID.
	CB_INVALID	To render the workflow invalid, set this parameter to ABAP_TRUE.

AFTER_START method

Calling sequence

This method is executed for all workflows that are pending for the document.

Method signature

Type	Parameter	Description
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data
	IR_WORKFLOW	Reference to the workflow object
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type and workflow ID.

AFTER_FINISH method

Calling sequence

This method is only called when the workflow is finished (that is, the last step of the workflow has been approved).

Method signature

Type	Parameter	Description
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data
	IR_WORKFLOW	Reference to the workflow object
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type and workflow ID.

BAdI Workflow steps handling

User exits / BAdIs > Platform > Workflow

This user exit / BAdI currently provides the interface methods CHECK_RECEIVER_VALIDITY, CHECK_STEP_VALIDITY, USER_DETERMINATION, AFTER_FINISHED_STEP, GET_ITEM_APPROVER and CHECK_ACTION that you can implement to add your own customer logic for handling workflow steps.

This user exit / BAdI is filter-dependent and allows you to filter on object (process type), workflow and workflow step.

Explanation of terms

Term	Explanation
Current steps	Steps for which the state is **Active**, that is, a processor is currently working on them.
Next steps	Steps whose position in the workflow definition is the same or greater than the position of the current step. In the case of steps with the same position as the current step, only those steps for which step processing has not yet started are taken into account. If the workflow has not yet started, all steps are considered as next steps.
Previous step(s)	Steps for which the state is **Done**. Invalid steps are not considered as previous steps.
Step being processed by the user / Steps being processed by the BAdI	This differentiation is due to the architecture used for the BAdI implementation. The *step being processed by the user* refers to the step that the user is currently processing in the PROCESS DIRECTOR application. When the user performs an action that triggers the BAdI, for some methods all steps of the workflow must be processed. This is performed in a loop, one step at time. In this case, the PROCESS DIRECTOR application processes not only steps that are being processed by the user, but also other steps in the workflow - these are then *steps being processed by the BAdI*.

CHECK_RECEIVER_VALIDITY method

This method can be used, for example, to determine whether a user is valid for a specific workflow action or not.

Calling sequence

This method is called in the following cases:

- For the APPROVE action – for next steps only
- For the CONTINUE action – for current and next steps
- For the WORKFLOW START action – for current and next steps
- For the FORWARD action – only for the step currently being processed by the user
- For the QUERY action – only for the step currently being processed by the user

Method signature

Type	Parameter	Remarks
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data
	IC_STEP_ID	Workflow step ID. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_CONF	Structure containing workflow step configuration. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_DATA	Structure containing workflow step data. This parameter refers to the step that is being processed by the BAdI method, by the user.
	IN_STEP_POSITION	This value represents the position of the step that is currently being processed by the user. **Note**: When the workflow is in the start phase but has not yet started, the value is set to 0. The same value is set when the workflow has already finished.
	IT_STEP_USER_TYPES	User types available for the step. This parameter refers to the step that is being processed by the BAdI method, not by the user.

Type	Parameter	Remarks
	IT_STEP_PROCESSORS	List containing all processors of the current and next steps
	IT_STEP_APPROVERS	List containing all step approvers of the current and previous steps
	IT_NEXT_STEPS	List containing all next steps with their details
	IT_CURR_STEPS	List containing all current steps with their details
	IT_PREV_STEPS	List containing all previous steps with their details
	IS_RECEIVER	The step recipient is passed here
	IR_STEP	Reference to the step object
	IR_RECEIVER	Reference to the step receiver object
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type, workflow ID and step ID.
	CB_INVALID	To render the recipient invalid, set this parameter to ABAP_TRUE.

CHECK_STEP_VALIDITY method

This method can be used, for example, to determine whether a step is valid or not.

Calling sequence

This method is called in the following cases:

- For the APPROVE action – for next steps only
- For the CONTINUE action – for current and next steps
- For the WORKFLOW START action – for current and next steps

This method is called *after* workflow step conditions (if any) have been evaluated for the given step.

Method signature

Type	Parameter	Remarks
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data
	IC_STEP_ID	Workflow step ID. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_CONF	Structure containing workflow step configuration. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_DATA	Structure containing workflow step data. This parameter refers to the step that is being processed by the BAdI method, by the user.

Type	Parameter	Remarks
	IN_STEP_POSITION	This value represents the position of the step that is currently being processed by the user. **Note**: When the workflow is in the start phase but has not yet started, the value is set to 0. The same value is set when the workflow has already finished.
	IT_STEP_USER_TYPES	User types available for the step. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IT_STEP_PROCESSORS	List containing all processors of the current and next steps
	IT_STEP_APPROVERS	List containing all step approvers of the current and previous steps
	IT_NEXT_STEPS	List containing all next steps with their details
	IT_CURR_STEPS	List containing all current steps with their details
	IT_PREV_STEPS	List containing all previous steps with their details
	IS_RECEIVER	The step recipient is passed here
	IR_STEP	Reference to the step object
	IR_RECEIVER	Reference to the step receiver object
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context

Type	Parameter	Remarks
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type, workflow ID and step ID.
	CB_INVALID	To render the step invalid, set this parameter to ABAP_TRUE.

USER_DETERMINATION method

This method can be used, for example, to add or remove workflow step processors.

Calling sequence

This method is only called for valid steps. It is called in the following cases:

- For the APPROVE action – for next steps only
- For the CONTINUE action – for current and next steps
- For the WORKFLOW START action – for current and next steps
- For the FORWARD action – only for the step currently being processed by the user
- For the QUERY action – only for the step currently being processed by the user

Method signature

Type	Parameter	Description
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data

Type	Parameter	Description
	IC_STEP_ID	Workflow step ID. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_CONF	Structure containing workflow step configuration. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_DATA	Structure containing workflow step data. This parameter refers to the step that is being processed by the BAdI method, by the user.
	IN_STEP_POSITION	This value represents the position of the step that is currently being processed by the user. **Note**: When the workflow is in the start phase but has not yet started, the value is set to 0. The same value is set when the workflow has already finished.
	IT_STEP_USER_TYPES	User types available for the step. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IT_STEP_PROCESSORS	List containing all processors of the current and next steps
	IT_STEP_APPROVERS	List containing all step approvers of the current and previous steps
	IT_NEXT_STEPS	List containing all next steps with their details
	IT_CURR_STEPS	List containing all current steps with their details
	IT_PREV_STEPS	List containing all previous steps with their details
	IR_STEP	Reference to the step object

Type	Parameter	Description
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type, workflow ID and step ID.
	CT_USERS	List of user IDs

AFTER_FINISHED_STEP method

Calling sequence

This method is called for all steps whose status is **Done** for the actions APPROVE, RECALL COMPLETE, RECALL PROCESSOR, REJECT, REPLY.

This method is called *after* a step is approved, rejected or recalled. If the step is a final step, this method is called before the **Workflow handling** user exit / BAdI method AFTER_FINISH.

Method signature

Type	Parameter	Remarks
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data

Type	Parameter	Remarks
	IC_STEP_ID	Workflow step ID. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_CONF	Structure containing workflow step configuration. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_DATA	Structure containing workflow step data. This parameter refers to the step that is being processed by the BAdI method, by the user.
	IN_STEP_POSITION	This value represents the position of the step that is currently being processed by the user. **Note**: When the workflow is in the start phase but has not yet started, the value is set to 0. The same value is set when the workflow has already finished.
	IT_STEP_USER_TYPES	User types available for the step. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IT_STEP_PROCESSORS	List containing all processors of the current and next steps
	IT_STEP_APPROVERS	List containing all step approvers of the current and previous steps
	IT_NEXT_STEPS	List containing all next steps with their details
	IT_CURR_STEPS	List containing all current steps with their details
	IT_PREV_STEPS	List containing all previous steps with their details

Type	Parameter	Remarks
	IR_STEP	Reference to the step object
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type, workflow ID and step ID.

GET_ITEM_APPROVER method

This method can be used, for example, to assign processors for line item approval. The approver is returned using the CS_APPROVER parameter and should be one of the step processors, who are available in the IT_LINE_PROCESSORS table.

Calling sequence

This method is called for active steps for which user assignment is necessary and line item approval is enabled. Each item of the document is processed in a loop. This method is called on workflow start and step approval.

Method signature

Type	Parameter	Description
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data

Type	Parameter	Description
	IC_STEP_ID	Workflow step ID. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_CONF	Structure containing workflow step configuration. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_DATA	Structure containing workflow step data. This parameter refers to the step that is being processed by the BAdI method, by the user.
	IN_STEP_POSITION	This value represents the position of the step that is currently being processed by the user. **Note**: When the workflow is in the start phase but has not yet started, the value is set to 0. The same value is set when the workflow has already finished.
	IT_STEP_USER_TYPES	User types available for the step. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IT_STEP_PROCESSORS	List containing all processors of the current and next steps
	IT_STEP_APPROVERS	List containing all step approvers of the current and previous steps
	IT_NEXT_STEPS	List containing all next steps with their details
	IT_CURR_STEPS	List containing all current steps with their details
	IT_PREV_STEPS	List containing all previous steps with their details

Type	Parameter	Description
	IT_LINE_PROCESSORS	Possible processors for the step
	IR_STEP	Reference to the step object
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type, workflow ID and step ID.
	CS_APPROVER	Approver data
	CB_PRESET_APPROVED	Preset the **Approved** flag

CHECK_ACTION method

Calling sequence

This method is called before any other workflow BAdI method. It is called during the approve, reject, forward and query actions.

Method signature

Type	Parameter	Remarks
	IC_ACTION	Action for which the BAdI is executed (domain /EBY/PDWC_MACTIVITY): ▪ APR - approve ▪ QRY - query ▪ FWD - forward ▪ RJC - reject
	IS_COORDINATES	Structure containing information about the current action chain, action and envelope
	IC_WC_ID	Workflow ID
	IS_WC_HDR	Structure containing workflow header data
	IS_WC_CONF	Structure containing workflow configuration
	IT_WC_ALL_STEPS	Table containing all workflow steps and their data
	IC_STEP_ID	Workflow step ID. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_CONF	Structure containing workflow step configuration. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IS_STEP_DATA	Structure containing workflow step data. This parameter refers to the step that is being processed by the BAdI method, by the user.

Type	Parameter	Remarks
	IN_STEP_POSITION	This value represents the position of the step that is currently being processed by the user. **Note**: When the workflow is in the start phase but has not yet started, the value is set to 0. The same value is set when the workflow has already finished.
	IT_STEP_USER_TYPES	User types available for the step. This parameter refers to the step that is being processed by the BAdI method, not by the user.
	IT_STEP_PROCESSORS	List containing all processors of the current and next steps
	IT_STEP_APPROVERS	List containing all step approvers of the current and previous steps
	IT_NEXT_STEPS	List containing all next steps with their details
	IT_CURR_STEPS	List containing all current steps with their details
	IT_PREV_STEPS	List containing all previous steps with their details
	IR_STEP	Reference to the step object
	IR_CURRENT_DOCUMENT	Reference to the current document
	IR_CONTEXT	Reference to the user session context
	VALUE(FLT_VAL)	Filter values passed implicitly to the BAdI and explicitly to the method. Structure contains object type, workflow ID and step ID.

Type	Parameter	Remarks
	EB_CANCEL	To render the action on a step invalid, you can set this parameter to ABAP_TRUE.
	ET_MESSAGES	List of messages to be added to the queue. Message of type 'E' (error) has the same meaning as the parameter EB_CANCEL set to ABAP_TRUE.

Process types reference

This section provides reference information for the following process types:

Accounts Payable (IV)

SAP Object type: `/EBY/PDIV`

Package: `/EBY/ICIV`

This chapter provides reference information for Accounts Payable. For detailed information on configuring Accounts Payable in PROCESS DIRECTOR, see the *PROCESS DIRECTOR Accounts Payable Configuration Guide*.

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

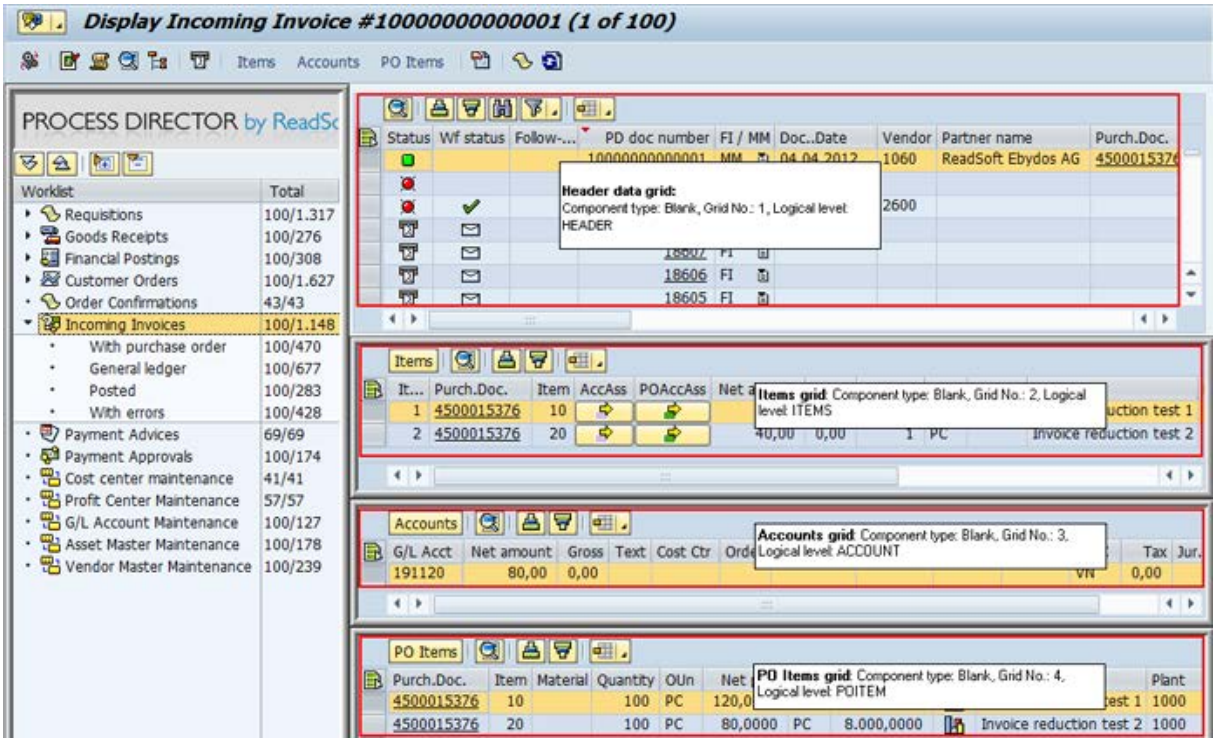
Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/ICIV_MODEL	/EBY/ICIV_SHDR_IF	CI_COCKPIT_HDR
Item data	/EBY/ICIV_MODEL	/EBY/ICIV_SITM_IF	CI_COCKPIT_ITEM
Account assignment data	/EBY/ICIV_MODEL	/EBY/ICIV_SPOACCASS_IF	
Tax data	/EBY/ICIV_MODEL	/EBY/ICIV_STAX_IF	CI_COCKPIT_TAX

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Accounts Payable documents. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

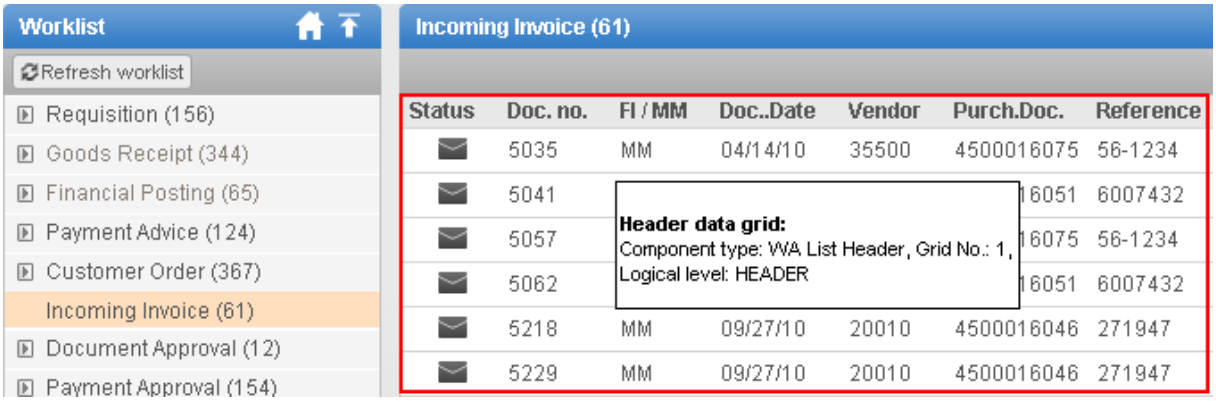
- SAP document overview list:



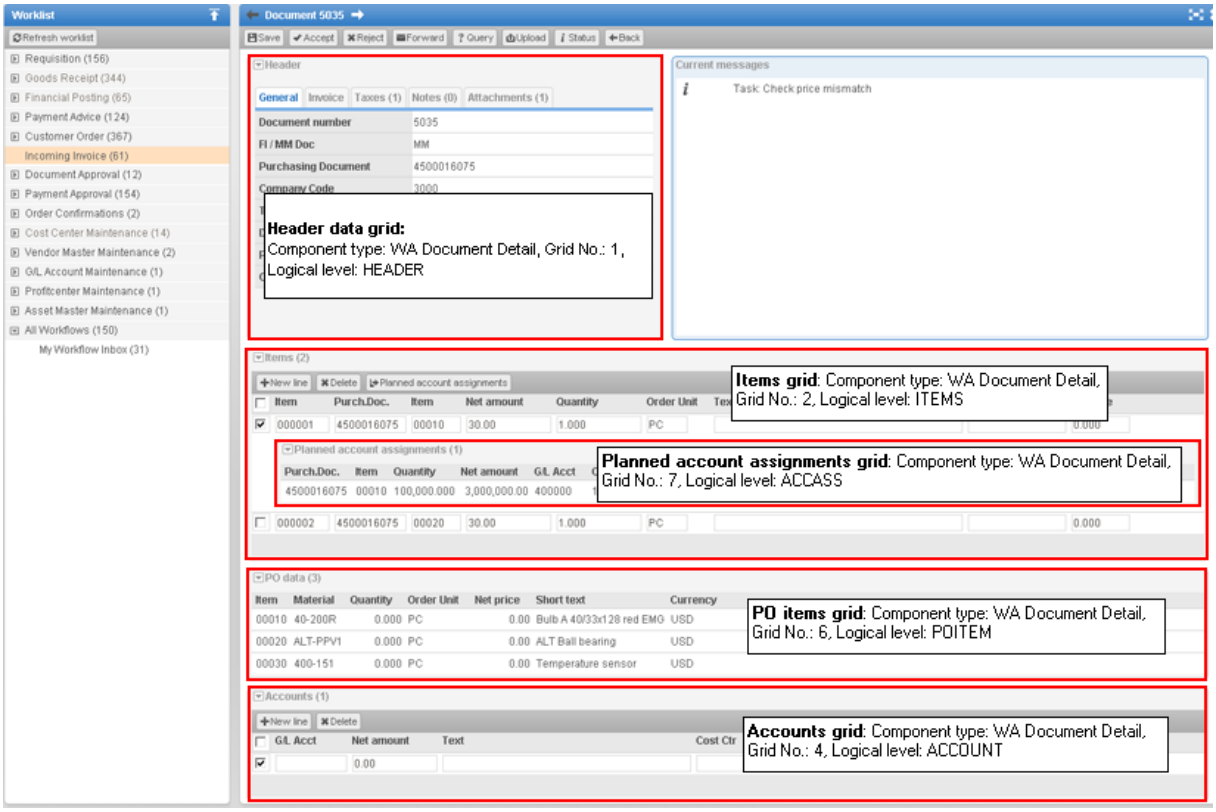
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

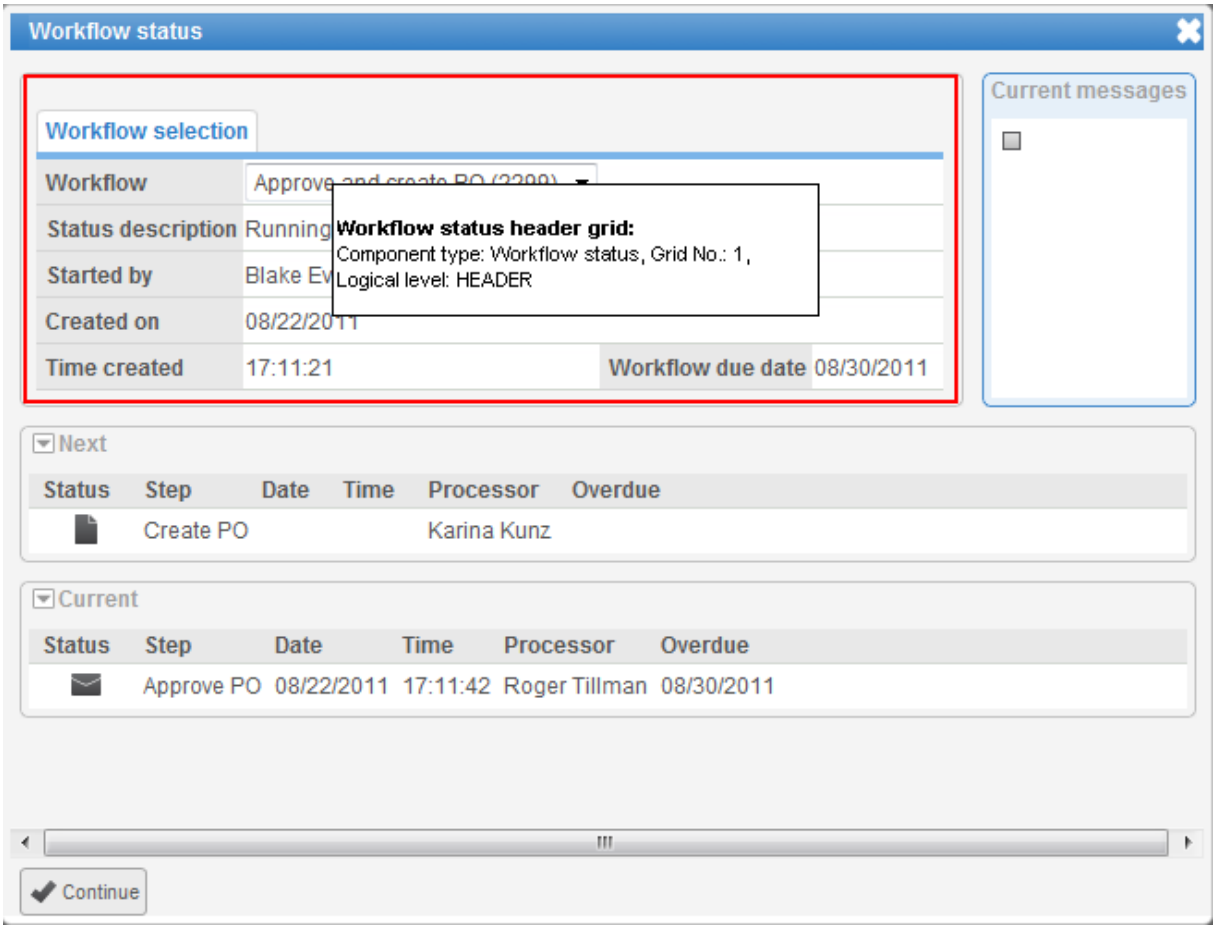
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDIV_SGRID1 ZXY_PDIV_LGRID1	Header Header data
	2	ITEMS	ZXY_PDIV_SGRID2 ZXY_PDIV_LGRID2	Items Items
	3	ACCOUNT	ZXY_PDIV_SGRID4 ZXY_PDIV_LGRID4	Accounts Accounts
	4	POITEM	ZXY_PDIV_SGRID6 ZXY_PDIV_LGRID6	PO items PO items

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
A1 Additional grid 01	1	TAX	ZXY_PDIV_SGRID3 ZXY_PDIV_LGRID3	Taxes Taxes
A1 Additional grid 01	2	ACCASS	ZXY_PDIV_SGRID5 ZXY_PDIV_LGRID5	Account assignments Account assignments
A1 Additional grid 01	4	PO_ACCASS	ZXY_PDIV_SA1_GRID5 ZXY_PDIV_LA1_GRID5	Planned account assignments Planned account assignments
WA Document Detail	1	HEADER	ZXY_PDIV_SWA_DOC_GRID1 ZXY_PDIV_LWA_DOC_GRID1	Remittance Advice Remittance Advice
WA Document Detail	2	ITEM	ZXY_PDIV_SWA_DOC_GRID2 ZXY_PDIV_LWA_DOC_GRID2	Items Items
WA Document Detail	3	TAX	ZXY_PDIV_SWA_DOC_GRID3 ZXY_PDIV_LWA_DOC_GRID3	Items Items
WA Document Detail	4	ACCOUNT	ZXY_PDIV_SWA_DOC_GRID4 ZXY_PDIV_LWA_DOC_GRID4	Items Items
WA Document Detail	5	ACCASS	ZXY_PDIV_SWA_DOC_GRID5 ZXY_PDIV_LWA_DOC_GRID5	Items Items
WA Document Detail	6	SORDER	ZXY_PDIV_SWA_DOC_GRID6 ZXY_PDIV_LWA_DOC_GRID6	Items Items
WA Document Detail	7	PO_ACCASS	ZXY_PDIV_SWA_DOC_GRID8 ZXY_PDIV_LWA_DOC_GRID8	Items Items
WA List Header	1	HEADER	ZXY_PDIV_SWA_LIST_GRID1 ZXY_PDIV_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDIV_SWA_STAT_GRID1 ZXY_PDIV_LWA_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDIV_SARC_LOG_GRID1 ZXY_PDIV_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses for Accounts Payable:

Status	Status Description
01	Unprocessed
02	Erroneous
03	Posted
04	Deleted
05	Partially posted
06	In workflow
07	WEB CYCLE for information
08	WEB CYCLE on posted documents
09	Paid
10	Collective invoice completed
11	Collective invoice uncompleted
13	Rejected
23	Parked
B1	New document in WEB BOARD
B2	Deleted document in WEB BOARD
F1	Fast entry

Text types

PROCESS DIRECTOR ships with this system text type for Accounts Payable:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

Programs

The following programs are available for use with Accounts Payable:

- `/EBY/ICIV_DOC_SYNCHRONIZE`

  Registers existing Accounts Payable (former INVOICE COCKPIT) documents in PROCESS DIRECTOR.

- `/EBY/ICIV_ACCRUAL_REPORT`

  Collects aggregated accrual information from all un-posted PROCESS DIRECTOR Accounts Payable documents.

/EBY/ICIV_DOC_SYNCHRONIZE

Use the `/EBY/ICIV_DOC_SYNCHRONIZE` program to register existing Accounts Payable (former INVOICE COCKPIT) documents in PROCESS DIRECTOR. It is necessary to run this program in order to be able to view and access Accounts Payable documents in PROCESS DIRECTOR.

You can specify which documents should be registered. For example, by default, documents that have already been completed in PROCESS DIRECTOR are excluded from registration, but you may want to include them.

Setting	Description
Identification	Unique document identifier (GUID).
Document number	PROCESS DIRECTOR document number.
PROCESS DIRECTOR doc. type	PROCESS DIRECTOR document type (defined in `/COCKPIT/C11`).

Setting	Description
Document date	Document date of the document (`DOC_DATE`).
FI / MM doc	Whether the document is an FI document or an MM document.
Include finished documents	Check to include documents that have already been processed and completed in PROCESS DIRECTOR (posted, parked or rejected documents and completed collective invoices).
System ID	When using the umbrella solution, specify the system ID of the remote system for which the documents should be synchronized to the umbrella system.

/EBY/ICIV_ACCRUAL_REPORT

Use this report to collect aggregated accrual information from all un-posted PROCESS DIRECTOR Accounts Payable documents. The report displays all PROCESS DIRECTOR Accounts Payable line items that have not yet been posted. You can select line items in the report and create a PROCESS DIRECTOR Financial Posting document consisting of all selected line items and one aggregated offsetting / balancing position. The report can also be run with transaction `/EBY/ICIV_ACCR_REP`.

See the *PROCESS DIRECTOR Accounts Payable User Guide* for detailed information on using this report.

Asset Acquisitions (AA)

SAP Object type: `/EBY/PDAA`

Package: `/EBY/PDAA`

This chapter provides reference information for Asset Acquisitions.

Checks for Asset Acquisitions

The following check is available for Asset Acquisitions in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check Asset Acquisition posting

`/EBY/CL_PDAA_EVT_CHK_POST`

Simulates the creation of an asset acquisition posting in SAP and returns all SAP messages generated during the simulation. Stores the result of the simulation for further analysis.

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

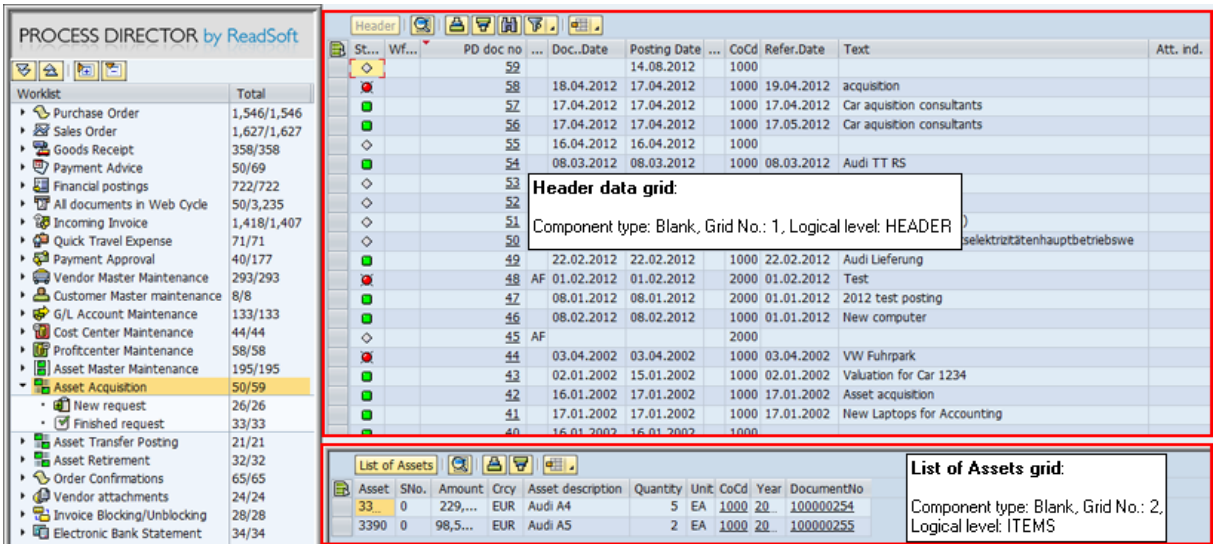
Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/PDAA_MODEL	/EBY/PDAA_SHDR_IF	CI_EBY_PDAA_SHDR_DATA CI_EBY_PDAA_SHDR_DISP
List of Assets data	/EBY/PDAA_MODEL	/EBY/PDAA_SLOA_IF	CI_EBY_PDAA_SLOA_DATA CI_EBY_PDAA_SLOA_DISP

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Asset Acquisitions. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

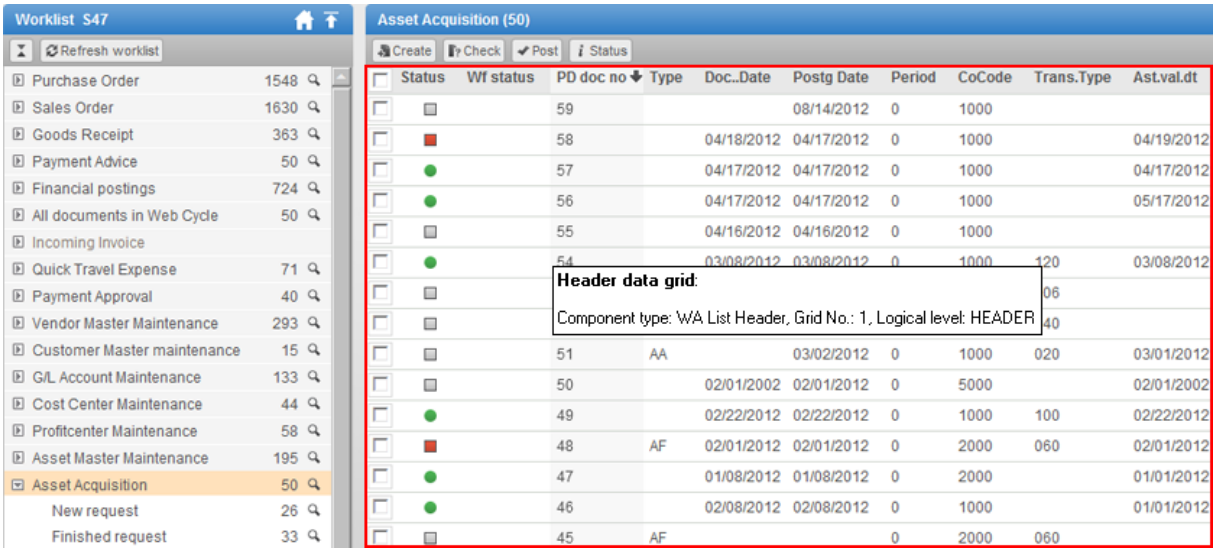
- SAP document overview list:



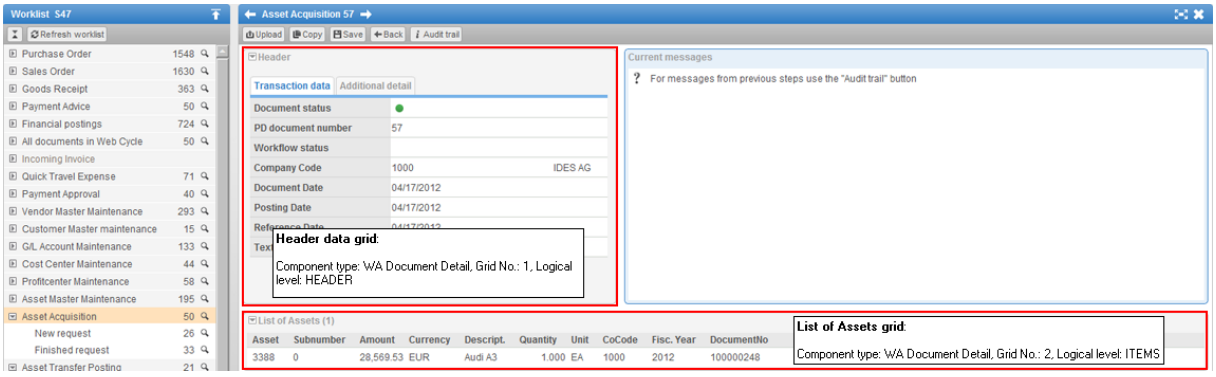
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

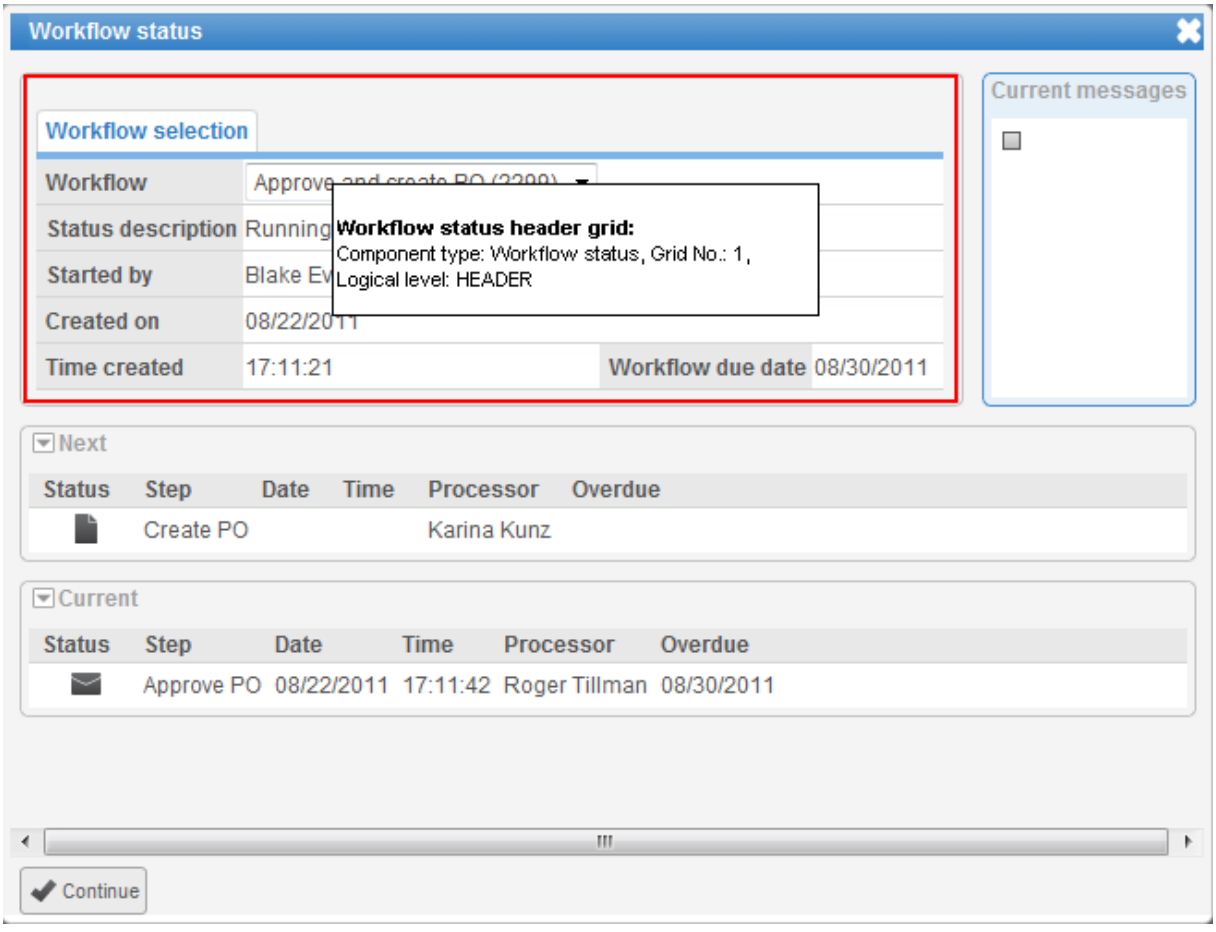
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

GUI area	Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming suggestions)	Btn. label Grid title (naming suggestions)
SAP document list		1	HEADER	ZXY_PDAA_SGRID1 ZXY_PDAA_LGRID1	Header Header data
SAP List of Assets grid		2	List of Assets	ZXY_PDAA_SGRID2 ZXY_PDAA_LGRID2	List of Assets List of Assets
Web Application document detail header data	WA Document Detail	1	HEADER	ZXY_PDAA_SWA_DOC_GRID1 ZXY_PDAA_LWA_DOC_GRID1	Header Header data
Web Application document detail List of Assets grid	WA Document Detail	2	List of Assets	ZXY_PDAA_SWA_DOC_GRID2 ZXY_PDAA_LWA_DOC_GRID2	List of Assets List of Assets
Web Application document list	WA List Header	1	HEADER	ZXY_PDAA_SWA_LIST_GRID1 ZXY_PDAA_LWA_LIST_GRID1	Header Header data
Workflow status popup	Workflow status	1	HEADER	ZXY_PDAA_SWA_STAT_GRID1 ZXY_PDAA_LWC_STAT_GRID1	Header Header data
Archive log	Header data for archive log	1	HEADER	ZXY_PDAA_SARC_LOG_GRID1 ZXY_PDAA_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses/substatuses for Asset Acquisitions:

Status	Status Description
ER	In error or incomplete
NN	New or ready for processing
OK	Processed
XX	Rejected

Text types

PROCESS DIRECTOR ships with this system text type for Asset Acquisitions:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

Asset Retirement (AR)

SAP Object type: /EBY/PDAR

Package: /EBY/PDAR

This chapter provides reference information for Asset Retirement.

Preset classes for Asset Retirement

The following preset class is available for Asset Retirement in the PROCESS DIRECTOR standard. You can also use general preset classes that are available for all process types.

Set substate based on transaction type

/EBY/CL_PDAR_PRESET_SUBSTATE

Sets the document substatus based on the transaction type.

Checks for Asset Retirement

The following check is available for asset retirement in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check Asset Retirement posting

/EBY/CL_PDAR_EVT_CHK_POST

Simulates the creation of an asset retirement posting in SAP and returns all SAP messages generated during the simulation. Stores the result of the simulation for further analysis.

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure,

you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

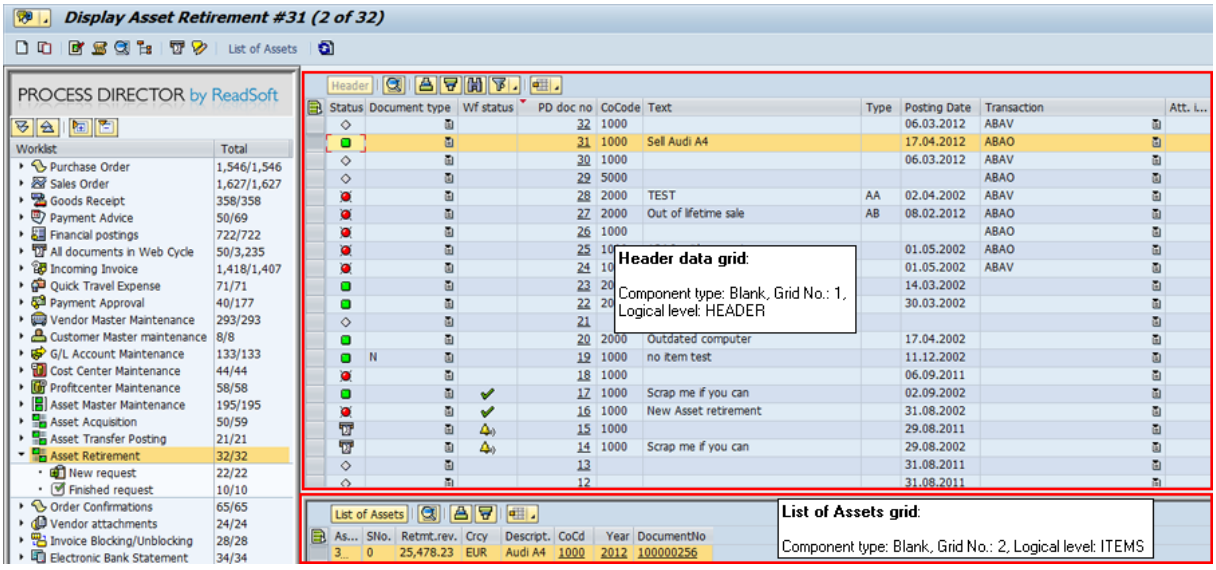
Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/PDAR_MODEL	/EBY/PDAR_SHDR_IF	CI_EBY_PDAR_SHDR_DATA CI_EBY_PDAR_SHDR_DISP
List of Assets data	/EBY/PDAR_MODEL	/EBY/PDAR_SLOA_IF	CI_EBY_PDAR_SLOA_DATA CI_EBY_PDAR_SLOA_DISP

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Asset Retirement. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

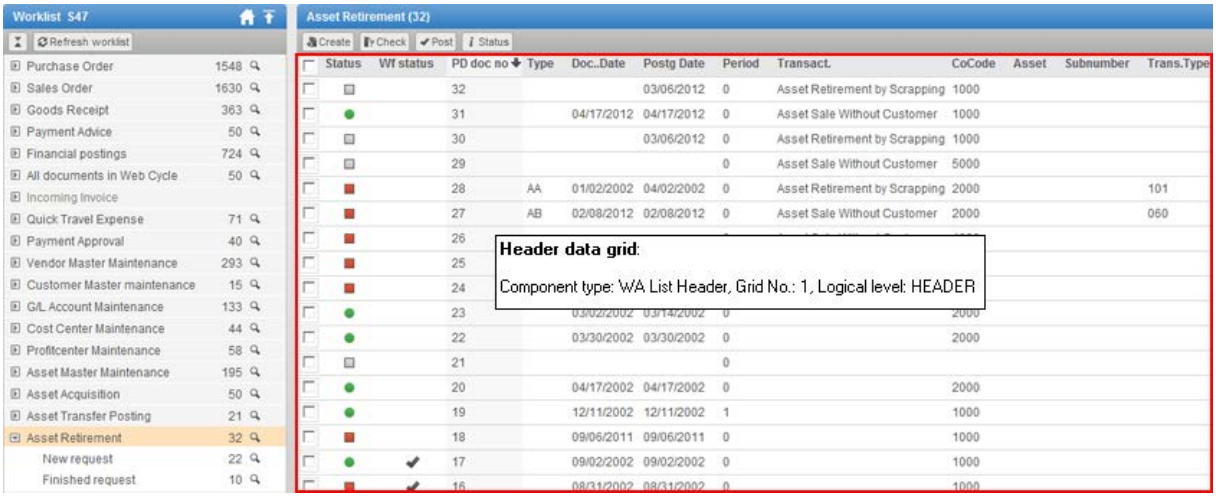
- SAP document overview list:



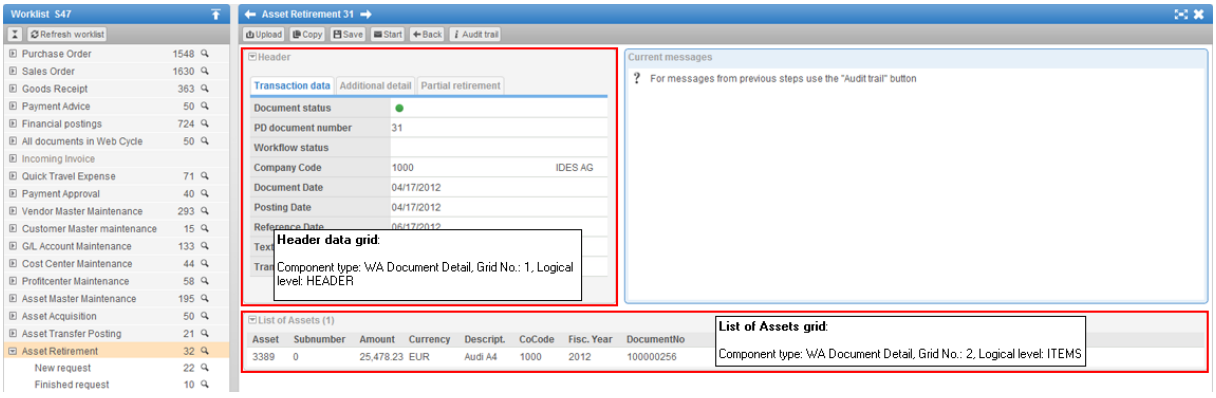
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

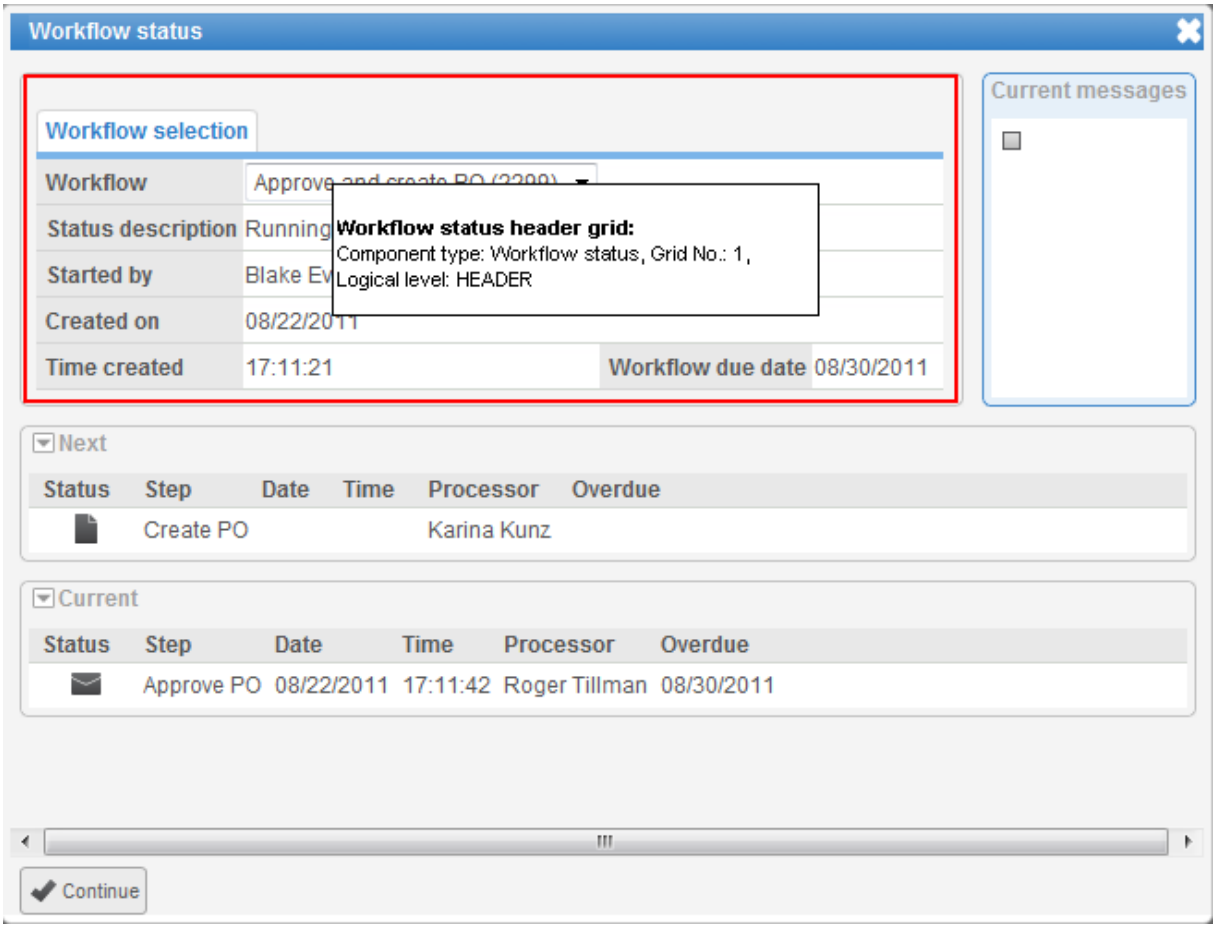
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

GUI area	Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming suggestions)	Btn. label Grid title (naming suggestions)
SAP document list		1	HEADER	ZXY_PDAR_SGRID1 ZXY_PDAR_LGRID1	Header Header data
SAP List of Assets grid		2	List of Assets	ZXY_PDAR_SGRID2 ZXY_PDAR_LGRID2	List of Assets List of Assets
Web Application document detail header data	WA Document Detail	1	HEADER	ZXY_PDAR_SWA_DOC_GRID1 ZXY_PDAR_LWA_DOC_GRID1	Header Header data
Web Application document detail List of Assets grid	WA Document Detail	2	List of Assets	ZXY_PDAR_SWA_DOC_GRID2 ZXY_PDAR_LWA_DOC_GRID2	List of Assets List of Assets
Web Application document list	WA List Header	1	HEADER	ZXY_PDAR_SWA_LIST_GRID1 ZXY_PDAR_LWA_LIST_GRID1	Header Header data
Workflow status popup	Workflow status	1	HEADER	ZXY_PDAR_SWA_STAT_GRID1 ZXY_PDAR_LWC_STAT_GRID1	Header Header data
Archive log	Header data for archive log	1	HEADER	ZXY_PDAR_SARC_LOG_GRID1 ZXY_PDAR_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses/substatuses for Asset Retirement:

Statuses

Status	Status Description
ER	In error or incomplete
NN	New or ready for processing
OK	Processed
XX	Rejected

The following system document substatuses are available for asset retirement:

Substatus	Description
BS	Asset Retirement by Scrapping
SW	Asset Sale Without Customer

Text types

PROCESS DIRECTOR ships with this system text type for Asset Retirement:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

Asset Transfer (AT)

SAP Object type: /EBY/PDAT

Package: /EBY/PDAT

This chapter provides reference information for Asset Transfer.

Preset classes for Asset Transfer

The following preset class is available for Asset Transfer in the PROCESS DIRECTOR standard. You can also use general preset classes that are available for all process types.

Set substate based on transaction type

/EBY/CL_PDAT_PRESET_SUBSTATE

Sets the document substatus based on the transaction type.

Checks for Asset Transfer

The following check is available for Asset Transfer in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check Asset Transfer posting

`/EBY/CL_PDAT_EVT_CHK_POST`

Simulates the creation of an asset transfer posting in SAP and returns all SAP messages generated during the simulation. Stores the result of the simulation for further analysis.

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

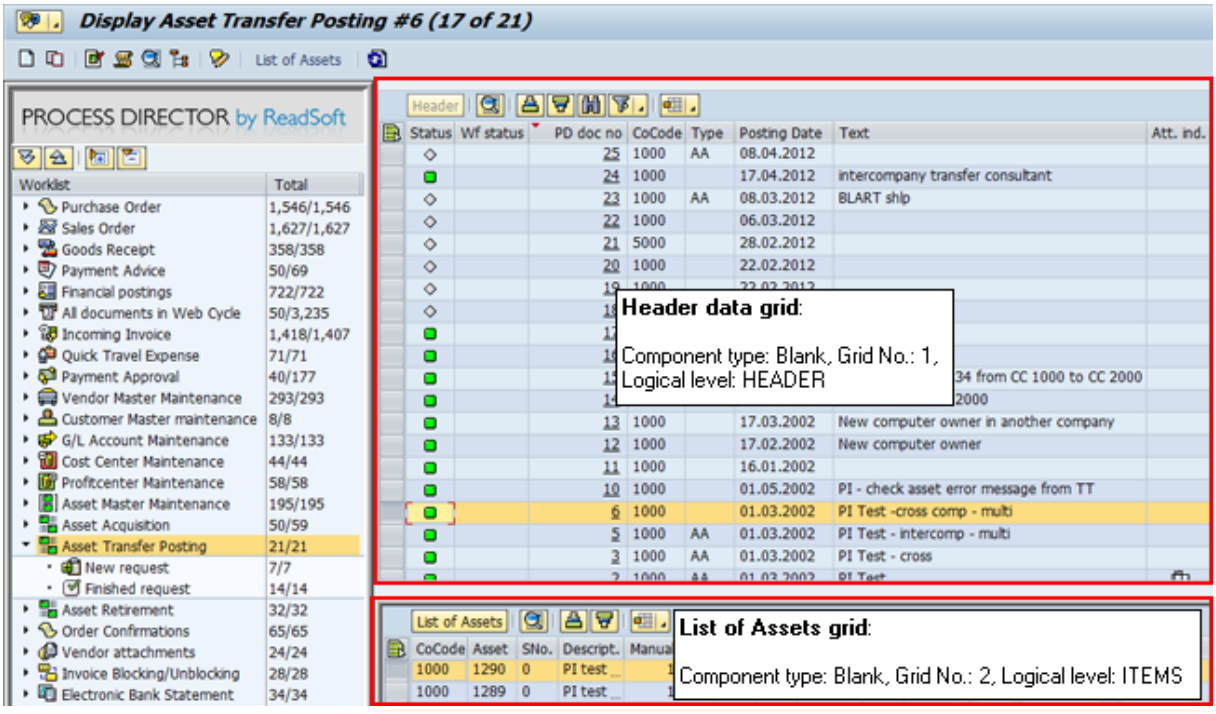
Logical level	Package	Interface structure	Customer Includes
Header data	`/EBY/PDAT_MODEL`	`/EBY/PDAT_SHDR_IF`	`CI_EBY_PDAT_SHDR_DATA` `CI_EBY_PDAT_SHDR_DISP`
List of Assets data	`/EBY/PDAT_MODEL`	`/EBY/PDAT_SLOA_IF`	`CI_EBY_PDAT_SLOA_DATA` `CI_EBY_PDAT_SLOA_DISP`

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Asset Transfer. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

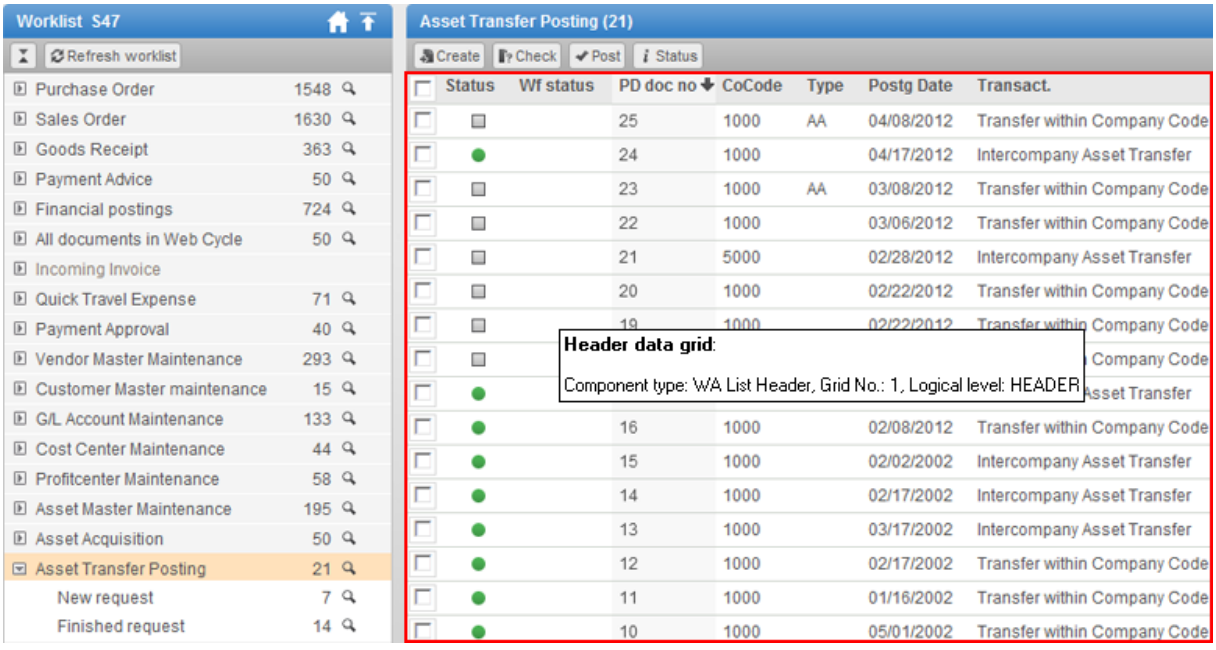
- SAP document overview list:



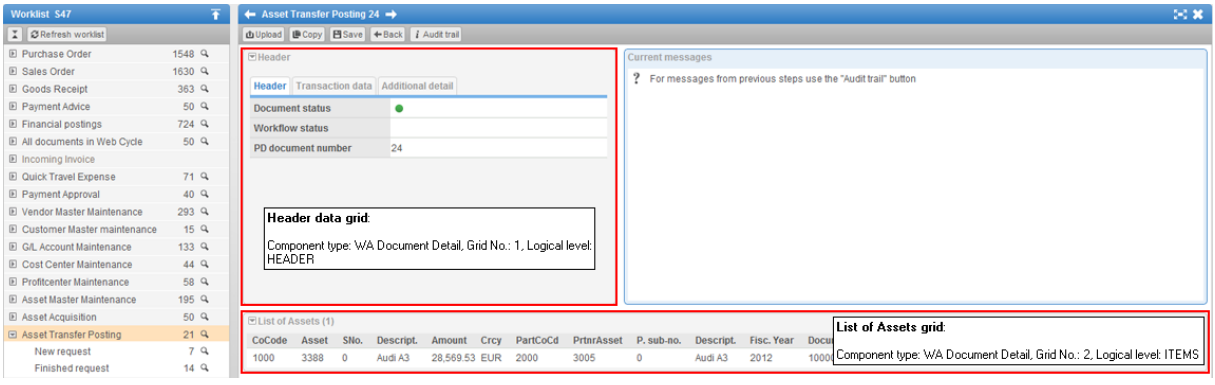
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

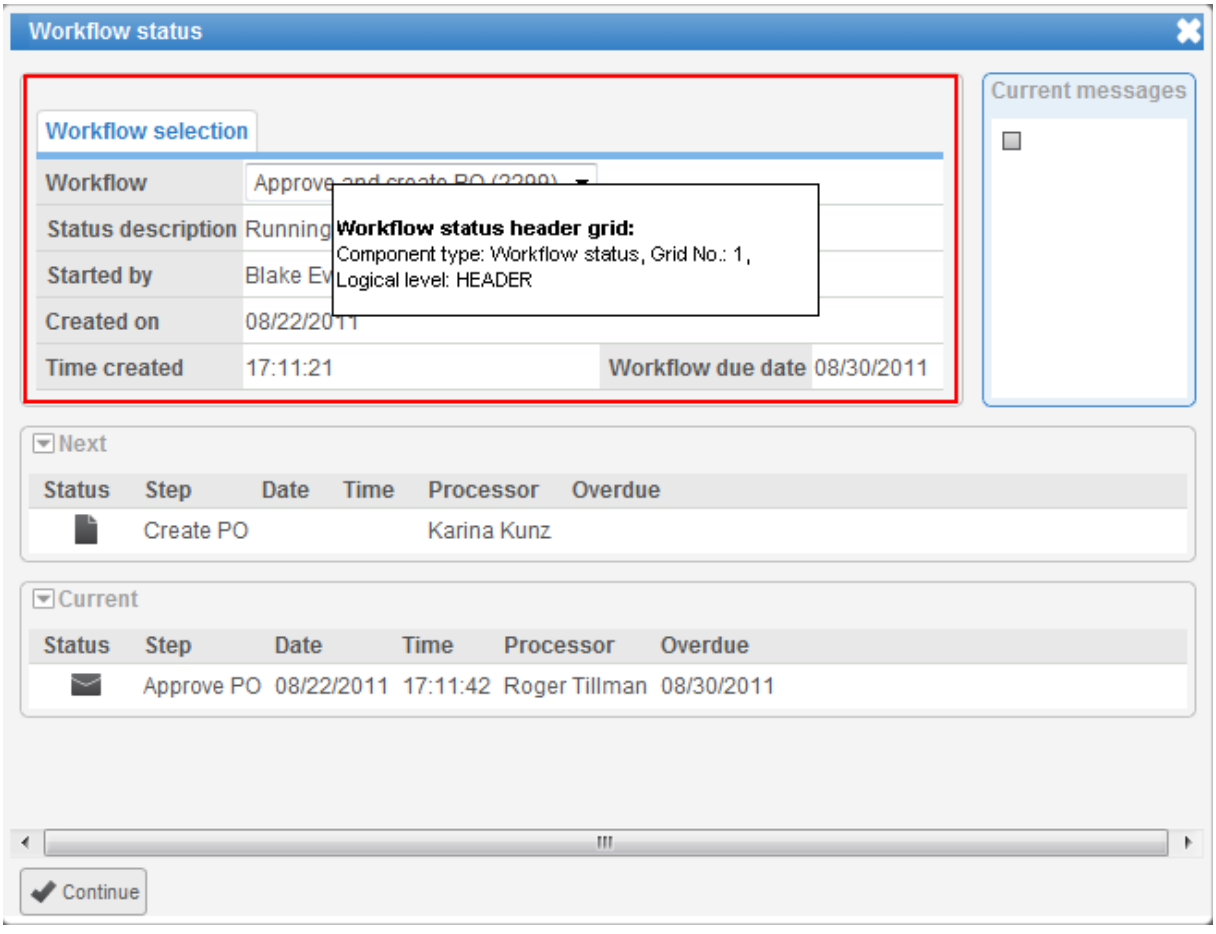
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

GUI area	Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming suggestions)	Btn. label Grid title (naming suggestions)
SAP document list		1	HEADER	ZXY_PDAT_SGRID1 ZXY_PDAT_LGRID1	Header Header data
SAP List of Assets grid		2	List of Assets	ZXY_PDAT_SGRID2 ZXY_PDAT_LGRID2	List of Assets List of Assets
Web Application document detail header data	WA Document Detail	1	HEADER	ZXY_PDAT_SWA_DOC_GRID1 ZXY_PDAT_LWA_DOC_GRID1	Header Header data

GUI area	Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming suggestions)	Btn. label Grid title (naming suggestions)
Web Application document detail List of Assets grid	WA Document Detail	2	List of Assets	ZXY_PDAT_SWA_DOC_GRID2 ZXY_PDAT_LWA_DOC_GRID2	List of Assets List of Assets
Web Application document list	WA List Header	1	HEADER	ZXY_PDAT_SWA_LIST_GRID1 ZXY_PDAT_LWA_LIST_GRID1	Header Header data
Workflow status popup	Workflow status	1	HEADER	ZXY_PDAT_SWA_STAT_GRID1 ZXY_PDAT_LWC_STAT_GRID1	Header Header data
Archive log	Header data for archive log	1	HEADER	ZXY_PDAT_SARC_LOG_GRID1 ZXY_PDAT_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses/substatuses for Asset Transfer:

Status	Status Description
ER	In error or incomplete
NN	New or ready for processing
OK	Processed
XX	Rejected

The following system document substatuses are available for asset transfer:

Substatus	Description
IC	Intercompany Asset Transfer
WI	Transfer within Company Code

Text types

PROCESS DIRECTOR ships with this system text type for Asset Transfer:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

Customer Orders (SO)

SAP Object type: /EBY/PDSO

Package: /EBY/PDSO

This chapter provides reference information for Customer Orders.

Checks for Customer Orders

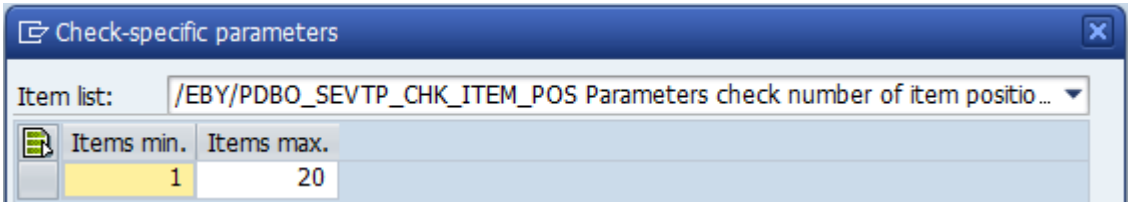
The following checks are available for Customer Orders in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check count of order item positions

/EBY/CL_PDSO_EVT_CHK_ITEMS

Checks for a required number of order line items and/or a maximum permitted number of line items. Also checks whether a material number has been entered for all line items.

Enter the minimum and maximum values in the initialization parameters.



For example, you can specify that a sales order must contain at least one line item and may have a maximum of 20 line items.

If one of these rules is broken, or if the material number is missing in any of the line items, a message is returned.

Check if order is in reference to another document

`/EBY/CL_PDSO_EVT_CHK_REFDOC`

Checks whether the order should be created in reference to another document because a reference document is mentioned.

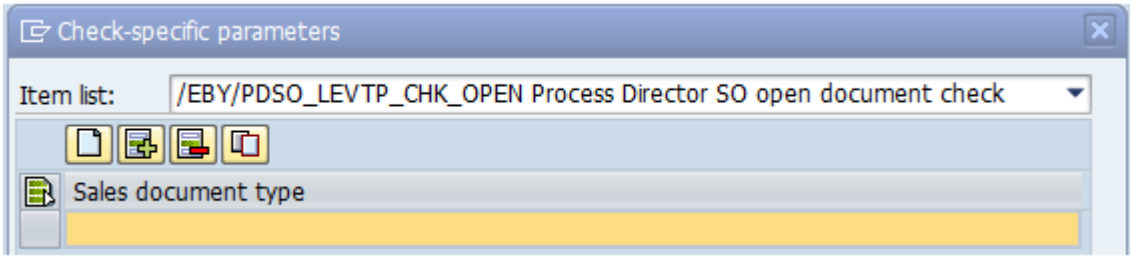
Note: If a reference could be a contract, ensure that the contract relation determination has been called to verify the contract reference.

Check if sold-to party has open documents

`/EBY/CL_PDSO_EVT_CHK_OPEN`

Checks whether an open contract or quotation exists for the given sold-to party and whether the **Reference doc.** field in the PROCESS DIRECTOR document is empty.

You can specify one or more sales document types in the initialization parameters.



This check is used to prevent automatic posting in systems where it is common to work with quotations, but the quotation number for the sales order has not been captured.

Check incompletion log

`/EBY/CL_PDSO_EVT_CHK_INCOMP`

Lists all entries from the incompletion log.

This check is dependent on the order simulation, so the **Check whether an SAP sales order can be created** check must have run successfully first.

Check incompletion log of created/simulated order for price

`/EBY/CL_PDSO_EVT_CHK_EDICOND`

Checks the incompletion log for entries specific to problems with the expected price and the simulated price.

This check is dependent on the order simulation, so the **Check whether an SAP sales order can be created** check must have run successfully first.

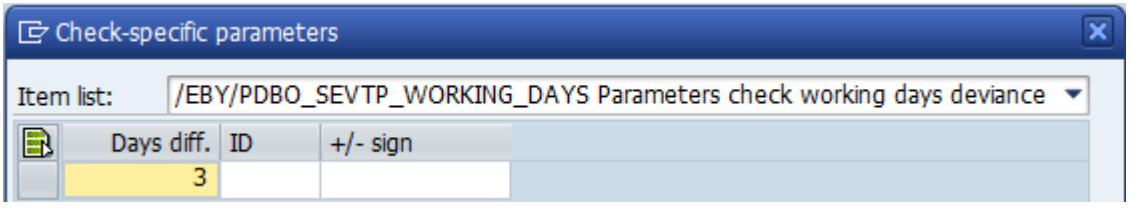
This check is only necessary if the condition types `EDI1` or `EDI2` are used in the calculation scheme and generate entries in the incompletion log.

Check item schedules regarding availability

`/EBY/CL_PDSO_EVT_CHK_SCHEDULES`

Checks calculated schedule dates on item level to indicate availability problems.

In the initialization parameters you can define allowed deviations from the target delivery date. If you also enter a factory calendar ID, the schedule dates calculation takes this calendar into account.

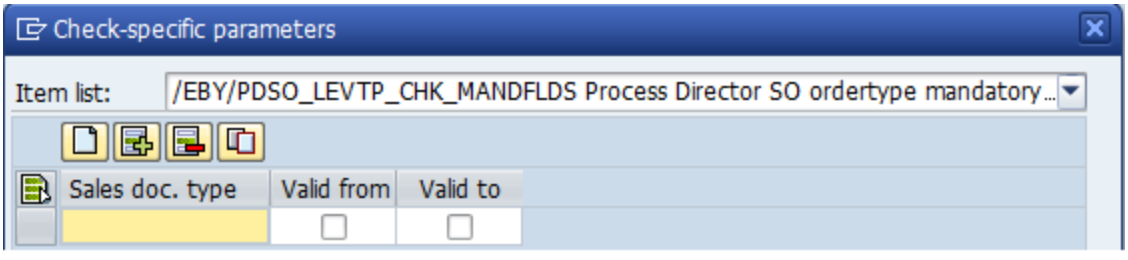


This check is dependent on the order simulation, so the **Check whether an SAP sales order can be created** check must have run successfully first.

Check mandatory fields for order type

`/EBY/CL_PDSO_EVT_CHK_MANDFLDS`

Checks mandatory fields for special order types. Specify the document types to check in the initialization parameters.



Examples:

- Quotations need a **Valid to** date.
- Scheduling agreements need a **Valid from** and a **Valid to** date.

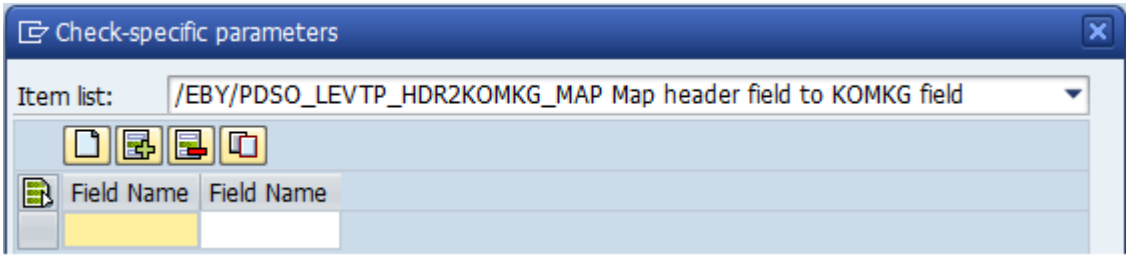
Check material of items against exclusion/list logic

`/EBY/CL_PDSO_EVT_CHK_MAT_EXCL`

Checks the Listing/Exclusion rules configured in SAP, which make it possible to allow or disallow the sale of specific materials to a specific customer.

This check returns an error message if a material that is excluded for the sold-to party is entered, or if the material is not included in the listed materials for the sold-to party.

By default, PROCESS DIRECTOR maps the sales organization, distribution channel, division, sales document type and partner number to the `KOMKG` structure. If additional header fields for Listing/Exclusion (such as customer-specific fields) are used in the `KOMKG` structure and should also be used for the check, these must be mapped to the corresponding PROCESS DIRECTOR fields. You can define this field mapping in the check's initialization parameters.



Check minimum order quantity for all items

`/EBY/CL_PDSO_EVT_CHK_MIN_ORD_Q`

Checks whether the minimum order quantity is reached for every line item.

Different order units are taken into account.

Check price discrepancies on item level

`/EBY/CL_PDSO_EVT_CHK_ITEMPRICE`

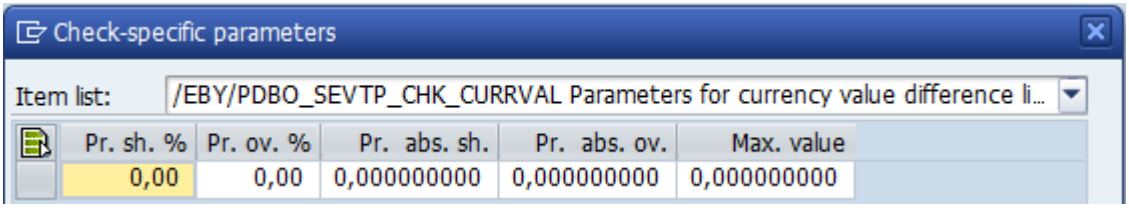
Checks price discrepancies on item level against allowed limits. Only possible for order items where a customer expected price is given.

In the initialization parameters, you can define the following rules:

- value shortfall in %
- value overrun in %
- absolute value shortfall
- absolute value overrun

- limit for standard line item value

If one of the defined rules is broken, a message is generated.



This check is dependent on the order simulation, so the **Check whether an SAP sales order can be created** check must have run successfully first.

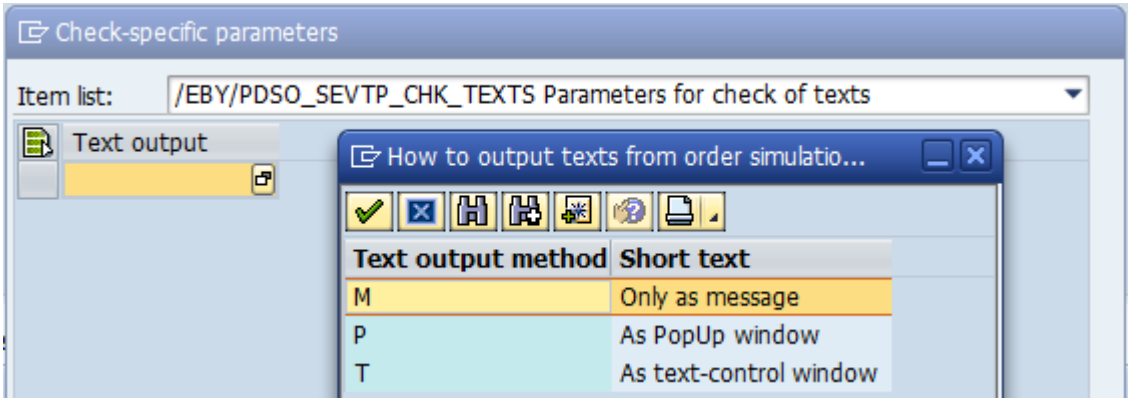
Check sales related texts

`/EBY/CL_PDSO_EVT_CHK_TEXTS`

Checks for sales-related texts, such as:

- customer information
- material information
- order type related information

In the initialization parameters you can define how the texts are displayed:



Initialization parameter	Description
M Only as message	Displays the text as a message. Suitable for background processing, but with the limitation that only 200 characters are displayed and there are no formatting options. This is the default setting.

Initialization parameter	Description
P As popup window	Displays the text in a popup, with the limitation that only the first 4 lines of the text are displayed.
T As text control window	Displays the entire text in the text editor.

Check whether an SAP sales order can be created

`/EBY/CL_PDSO_EVT_CHK_CREATE`

Simulates the creation of the sales order in SAP and returns all SAP messages generated during the simulation. Stores the result of the simulation for further analysis.

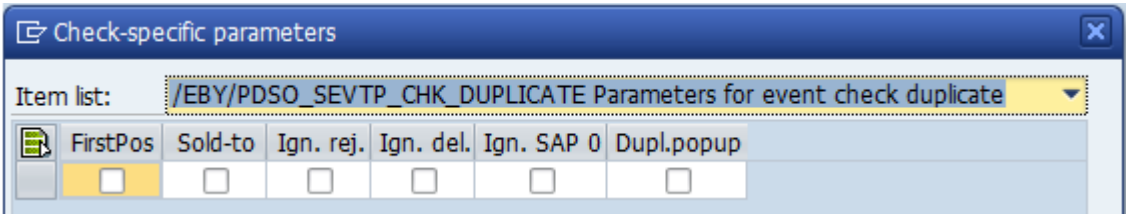
Check whether PO number has been already used

`/EBY/CL_PDSO_EVT_CHK_DUPLICATE`

Checks whether the number in the **PO number** field has already been used.

First all PROCESS DIRECTOR documents, then all posted SAP orders are checked.

You can specify these initialization parameters:



Initialization parameter	Description
FirstPos	Select to compare the PO number and also the first line item of the order with existing orders (item number, material and quantity are compared).
Sold-to	Select to compare the PO number and also the sold-to party customer number with existing orders.

Initialization parameter	Description
Ign. rej.	Select to ignore PROCESS DIRECTOR documents that have been rejected. These are not considered as duplicates even if the PO number is the same as the checked document.
Ign. del.	Select to ignore PROCESS DIRECTOR documents that have been deleted.
Ign. SAP 0	Select to ignore SAP documents that have a total header value equal to zero. These are not considered as duplicates even if the PO number is the same as the checked document.
Dupl.popup	Select to display a popup if a purchase order number that already exists is captured or entered, and the duplicate check is configured accordingly. The user can then view the documents that have the same PO number, and choose to reject the current or original document, manually update the sales order in SAP, or delete the PROCESS DIRECTOR document. See the *PROCESS DIRECTOR SAP User Guide* for more information.

Compare header net values of document with simulated order

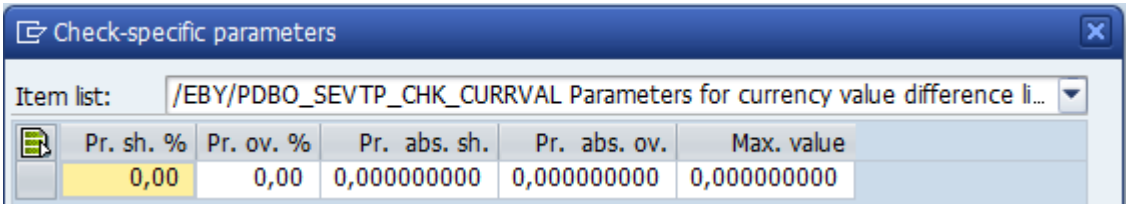
`/EBY/CL_PDSO_EVT_CHK_HDRNETVAL`

If an expected order value is given, this can be checked against defined ranges.

In the initialization parameters, you can define the following rules:

- value shortfall in %
- value overrun in %
- absolute value shortfall
- absolute value overrun
- limit for standard line item value

If one of the defined rules is broken, a message is generated.



This check is dependent on the order simulation, so the **Check whether an SAP sales order can be created** check must have run successfully first.

Test availability for every item position

`/EBY/CL_PDSO_EVT_CHK_ATP_QTY`

Checks the availability of every ordered line item by calling the SAP function module `BAPI_MATERIAL_AVAILABILITY` for the defined delivery plant.

Determinations for Customer Orders

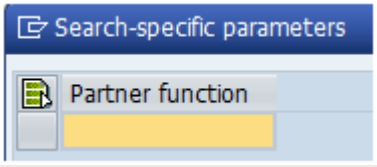
These determinations are available for Customer Orders in the PROCESS DIRECTOR standard:

Sales order BillTo determination

`/EBY/CL_PDSO_DET_BILL_TO_INIT`

Determines the Bill-To party of a customer order by searching for the following known information in customer master records (depending on the defined searches):

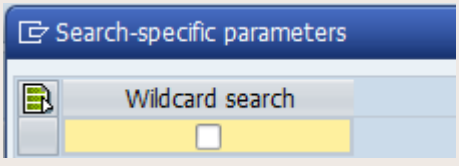
Search area	Search name
Customer number	`/EBY/CL_PDSO_DET_PARTNER_KNA1`
Customer account number	`/EBY/CL_PDSO_DET_PARTNER_KNB1` (finance) `/EBY/CL_PDSO_DET_PARTNER_KNVV` (sales)
Name or partial name	`/EBY/CL_PDSO_DET_PARTNER_NAME`

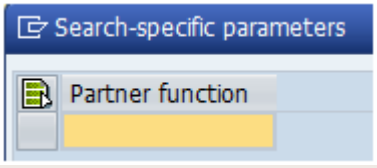
Search area	Search name
Address and contact data	`/EBY/CL_PDSO_DET_PARTNER_ZIP` (postal code) `/EBY/CL_PDSO_DET_PARTNER_ZIP1` (first part postal code) `/EBY/CL_PDSO_DET_PARTNER_ZIP` (city) `/EBY/CL_PDSO_DET_PARTNER_STR` (street name or partial street name) `/EBY/CL_PDSO_DET_PARTNER_TEL` (telephone number) `/EBY/CL_PDSO_DET_PARTNER_FAX` (fax number) `/EBY/CL_PDSO_DET_PARTNER_MAIL` (email address)
Tax numbers	`/EBY/CL_PDSO_DET_PARTNER_VAT` (VAT number) `/EBY/CL_PDSO_DET_PARTNER_TAX` (Tax number) `/EBY/CL_PDSO_DET_PARTNER_SIREN` (Tax number 2)
Business partners Specify the partner function in the search parameters: 	`/EBY/CL_PDSO_DET_PARTNER_SP` (Sold-To party) `/EBY/CL_PDSO_DET_PARTNER_SH` (Ship-To party) `/EBY/CL_PDSO_DET_PARTNER_CP` (Contact person)

Sales order Contact determination

`/EBY/CL_PDSO_DET_CONTACT_INIT`

Determines the contact person of a customer order by searching for the following known information in customer master records (depending on the defined searches):

Search area	Search name
Name or partial name In the search parameters, you can specify that wild cards can be used in the search. 	`/EBY/CL_PDSO_DET_CONTACT_NAME`

Search area	Search name
Name from the list of all maintained contacts of a given business partner (Sold-To, Ship-To, Bill-To). Specify the partner function in the search parameters: 	`/EBY/CL_PDSO_DET_CONTACT_P_N`
Contact data	`/EBY/CL_PDSO_DET_CONTACT_TEL` (telephone number) `/EBY/CL_PDSO_DET_CONTACT_FAX` (fax number) `/EBY/CL_PDSO_DET_CONTACT_MAIL` (email address)
Business partners	`/EBY/CL_PDSO_DET_CONTACT_SD` (Sold-To party) `/EBY/CL_PDSO_DET_CONTACT_SH` (Ship-To party)

Sales order contract relation determination

`/EBY/CL_PDSO_DET_CONTRACT_INIT`

Evaluates whether there are existing contracts for a sales order by (depending on the defined searches):

Search type	Search name
Reading all valid contracts of the customer	`/EBY/CL_PDSO_DET_CONTRACT_CUST`
Checking whether the customer purchase order is a contract reference	`/EBY/CL_PDSO_DET_CONTRACT_PO`

Search type	Search name
Checking whether the reference document is a contract reference	/EBY/CL_PDSO_DET_CONTRACT_REF

If the determination is successful, the sales order document type is switched to a contract and the contract is mentioned as the reference document on item level as well.

Sales order doc type determination

/EBY/CL_PDSO_DET_SO_TYP_INIT

Determines the correct sales document type using the search /EBY/CL_PDSO_DET_SO_TYP_DUMMY.

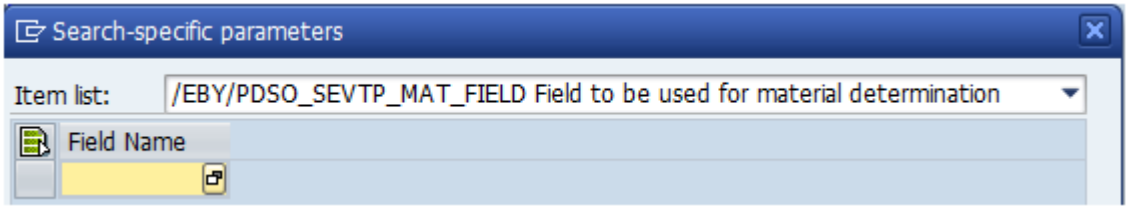
Sales order Material determination

/EBY/CL_PDSO_DET_MAT_INIT

Determines the material number by checking whether the material number given on the customer purchase order is (depending on the defined searches):

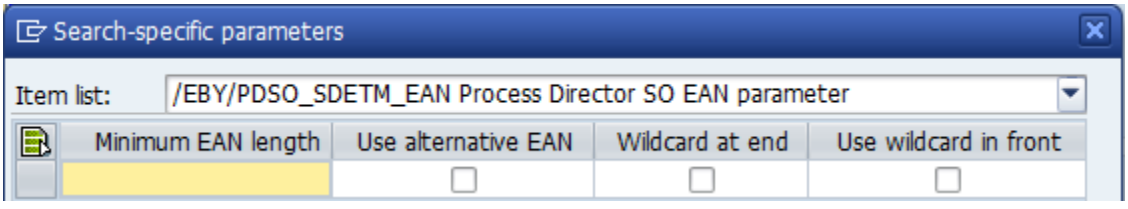
Search type	Search name
The material number from the material master data	/EBY/CL_PDSO_DET_MAT_MARA
A manufacturer part number maintained in the material master data	/EBY/CL_PDSO_DET_MAT_MANU
An EAN maintained in the material master data	/EBY/CL_PDSO_DET_MAT_EAN
A customer part number maintained in SAP table KNMT (transactions VD51/VD52/VD53)	/EBY/CL_PDSO_DET_MAT_CUST
The material number of a substitute material	/EBY/CL_PDSO_DET_MAT_SUBST

In the search parameters, you can specify which field should be used for material determination:



When searching by manufacturer or customer part number or EAN, you can specify a minimum length in the search parameters. The search will not be executed if the length of the material number given on the customer order is below this value.

For EANs, you can activate the search parameter **Alt. EAN** to search for alternative EANs (`MEAN` table) instead of main EAN numbers (`MARA` table).



Materials that are not valid for the given sales organization and materials that are blocked for sales are automatically eliminated from the search results.

If an exact match is not found, the searches use wild cards, first at the end and then at the beginning of the material number. Thus, a search for 2345 will find 12345 and 23456.

Sales order Plant determination

`/EBY/CL_PDSO_DET_PLANT_INIT`

Determines the correct delivery plant for a customer order line item by searching for this information in (depending on the defined searches):

Search type	Search name
The material master data	`/EBY/CL_PDSO_DET_PLANT_MAT`
The customer master data	`/EBY/CL_PDSO_DET_PLANT_SORG`

Sales order sales organization determination

`/EBY/CL_PDSO_DET_SORG_INIT`

Uses the search `/EBY/CL_PDSO_DET_SORG_SP` to determine the sales organization data for a customer order by retrieving it from the customer master data of the Sold-To party.

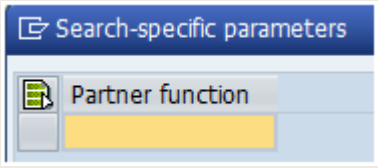
This determination can only succeed if the Sold-To party is known.

Sales order ShipTo determination

`EBY/CL_PDSO_DET_SHIP_TO_INIT`

Determines the Ship-To party of a customer order by searching for the following known information in customer master records (depending on the defined searches):

Search area	Search name
Customer number	`/EBY/CL_PDSO_DET_PARTNER_KNA1`
Customer account number	`/EBY/CL_PDSO_DET_PARTNER_KNB1` (finance) `/EBY/CL_PDSO_DET_PARTNER_KNVV` (sales)
Name or partial name	`/EBY/CL_PDSO_DET_PARTNER_NAME`
Address and contact data	`/EBY/CL_PDSO_DET_PARTNER_ZIP` (postal code) `/EBY/CL_PDSO_DET_PARTNER_ZIP1` (first part postal code) `/EBY/CL_PDSO_DET_PARTNER_ZIP` (city) `/EBY/CL_PDSO_DET_PARTNER_STR` (street name or partial street name) `/EBY/CL_PDSO_DET_PARTNER_TEL` (telephone number) `/EBY/CL_PDSO_DET_PARTNER_FAX` (fax number) `/EBY/CL_PDSO_DET_PARTNER_MAIL` (email address)
Tax numbers	`/EBY/CL_PDSO_DET_PARTNER_VAT` (VAT number) `/EBY/CL_PDSO_DET_PARTNER_TAX` (Tax number) `/EBY/CL_PDSO_DET_PARTNER_SIREN` (Tax number 2)

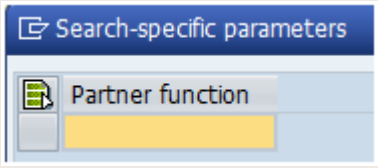
Search area	Search name
Business partners Specify the partner function in the search parameters: 	/EBY/CL_PDSO_DET_PARTNER_SP (Sold-To party) /EBY/CL_PDSO_DET_PARTNER_BP (Bill-To party) /EBY/CL_PDSO_DET_PARTNER_CP (Contact person) /EBY/CL_PDSO_DET_PARTNER_DEFSH (finds the default Ship-To party for the given Sold-To party)

Sales order SoldTo determination

/EBY/CL_PDSO_DET_SOLD_TO_INIT

Determines the Sold-To party of a customer order by searching for the following known information in customer master records (depending on the defined searches):

Search area	Search name
Customer number	/EBY/CL_PDSO_DET_PARTNER_KNA1
Customer account number	/EBY/CL_PDSO_DET_PARTNER_KNB1 (finance) /EBY/CL_PDSO_DET_PARTNER_KNVV (sales)
Name or partial name	/EBY/CL_PDSO_DET_PARTNER_NAME
Address and contact data	/EBY/CL_PDSO_DET_PARTNER_ZIP (postal code) /EBY/CL_PDSO_DET_PARTNER_ZIP1 (first part postal code) /EBY/CL_PDSO_DET_PARTNER_ZIP (city) /EBY/CL_PDSO_DET_PARTNER_STR (street name or partial street name) /EBY/CL_PDSO_DET_PARTNER_TEL (telephone number) /EBY/CL_PDSO_DET_PARTNER_FAX (fax number) /EBY/CL_PDSO_DET_PARTNER_MAIL (email address)
Tax numbers	/EBY/CL_PDSO_DET_PARTNER_VAT (VAT number) /EBY/CL_PDSO_DET_PARTNER_TAX (Tax number) /EBY/CL_PDSO_DET_PARTNER_SIREN (Tax number 2)

Search area	Search name
Business partners Specify the partner function in the search parameters: 	`/EBY/CL_PDSO_DET_PARTNER_SH` (Ship-To party) `/EBY/CL_PDSO_DET_PARTNER_BP` (Bill-To party) `/EBY/CL_PDSO_DET_PARTNER_CP` (Contact person)

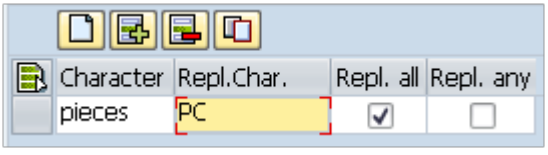
Sales order UOM determination

`/EBY/CL_PDSO_DET_UOM_INIT`

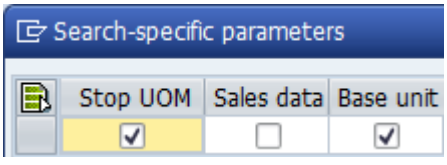
Determines the correct sales unit for a customer order line item by searching for this information in (depending on the defined searches):

Search type	Search name
The UoM description captured from the customer order	`/EBY/CL_PDSO_DET_UOM_EXT`
The material master data	`/EBY/CL_PDSO_DET_UOM_MASTER`

If a unit of measurement is captured, enter in the search parameters which captured UoM description should be converted to which SAP sales unit. For example, if the captured UoM description is `pieces`, enter this together with the SAP sales unit `PC` in the search parameters:

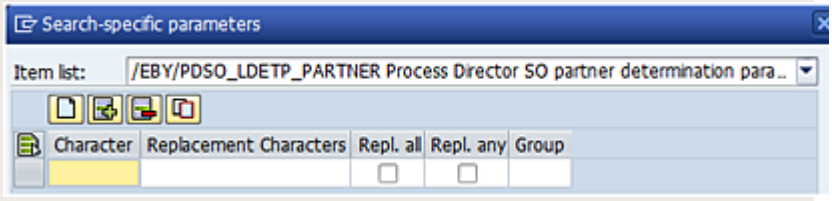
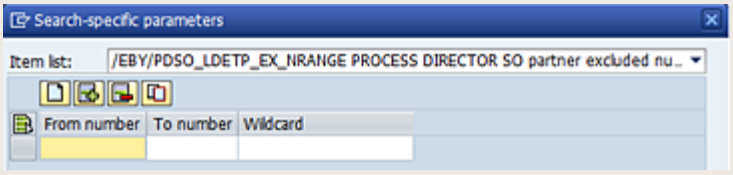


When searching the material master data, you can specify in the parameters whether to search only the base UoM in the material master (`MARA` table), only the UoM from the material sales data (`MVKE` table), or both. If you check the **Stop UOM** parameter, this search is not carried out if the unit of measurement has been captured.



Search parameters for partner searches

When performing partner searches (Bill-To, Ship-To, Sold-To determination), you can specify these parameters:

Parameter type	Description
/EBY/PDSO_LDETP_PARTNER	**PROCESS DIRECTOR SO partner determination parameters** Enables you to replace characters while searching. 
/EBY/PDSO_LDETP_EX_NRANGE	**PROCESS DIRECTOR SO partner excluded number range** Enables you to exclude a specific number range from the search. 

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

Logical level	Package	Interface structure	Customer Includes
Header data	`/EBY/PDSO_MODEL`	`/EBY/PDSO_SHDR_IF`	`CI_EBY_PDSO_SHDR_DATA` `CI_EBY_PDPO_SHDR_DISP`
Item data	`/EBY/PDSO_MODEL`	`/EBY/PDSO_SITM_IF`	`CI_EBY_PDSO_SITM_DATA` `CI_EBY_PDPO_SITM_DISP`
Conditions data	`/EBY/PDSO_MODEL`	`/EBY/PDSO_SCON_IF`	`CI_EBY_PDSO_SCON_DATA` `CI_EBY_PDPO_SCON_DISP`
Partner data	`/EBY/PDSO_MODEL`	`/EBY/PDSO_SPAR_IF`	`CI_EBY_PDPO_SPAR_DATA` `CI_EBY_PDPO_SPAR_DISP`
Schedule data	`/EBY/PDSO_MODEL`	`/EBY/PDSO_SSDL_IF`	`CI_EBY_PDPO_SSCH_DATA` `CI_EBY_PDPO_SSCH_DISP`

BAPI structure fields

Some document data models include BAPI structures. These structures can contain a large number of fields, many of which are not used. Therefore, most document data model BAPI structure fields are transient by default.

However, a customer may require that one or more of these fields be made persistent. PROCESS DIRECTOR offers an easy way to fulfill this requirement. In SAP transaction `SE80`, simply add the BAPI structure field(s) to the appropriate customer include.

Logical level	Package	Structure	Customer Include
Header data	`/EBY/PDSO_PERSISTENCE`	`/EBY/PDSO_THDR`	`CI_EBY_PDSO_SHDR_BAPI`

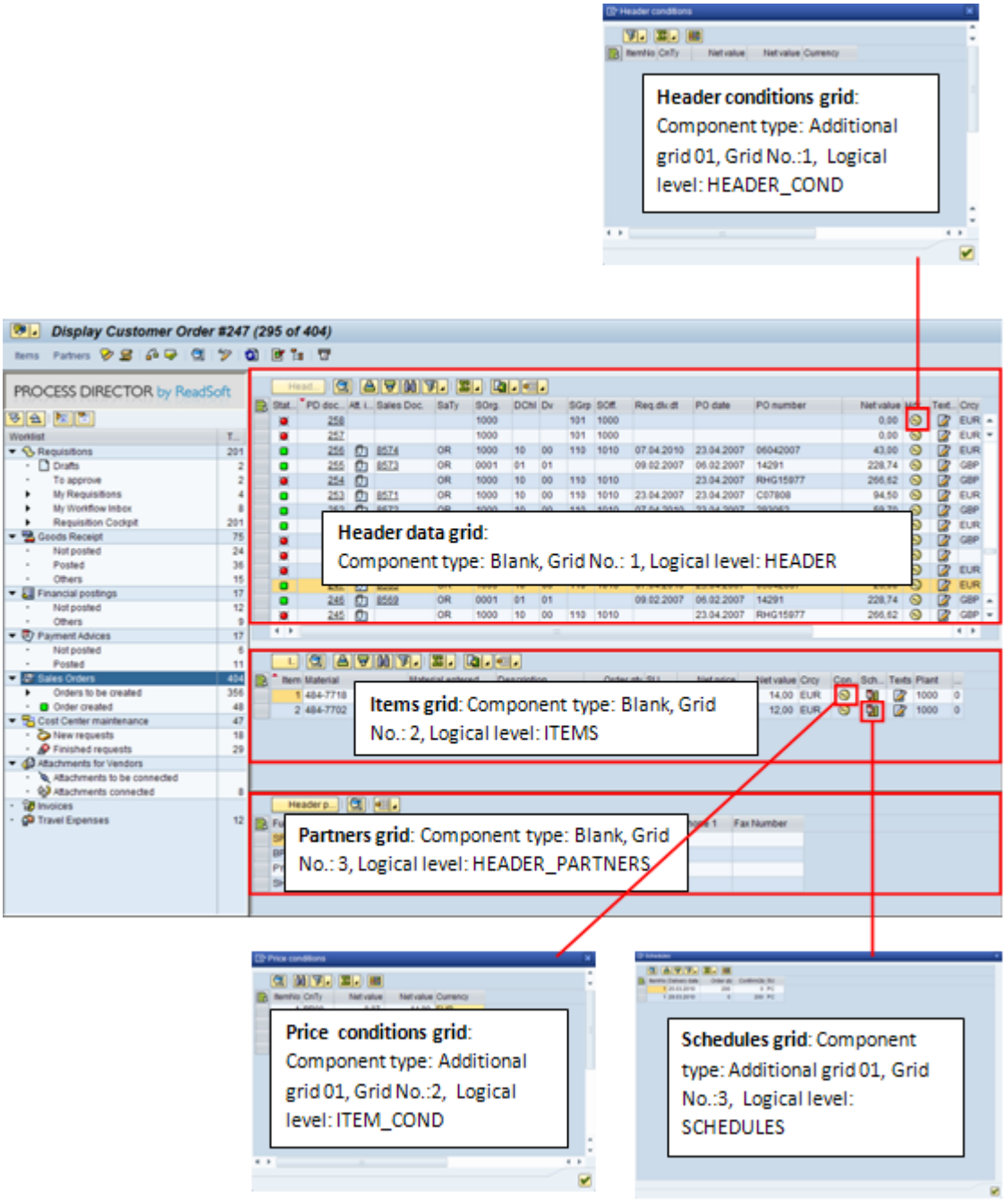
Logical level	Package	Structure	Customer Include
Item data	/EBY/PDSO_PERSISTENCE	/EBY/PDSO_TITM	CI_EBY_PDSO_SITM_BAPI
Conditions data	/EBY/PDSO_PERSISTENCE	/EBY/PDSO_TCON	CI_EBY_PDSO_SCON_BAPI
Partner data	/EBY/PDSO_PERSISTENCE	/EBY/PDSO_TPAR	CI_EBY_PDSO_SPAR_BAPI
Schedule data	/EBY/PDSO_PERSISTENCE	/EBY/PDSO_TSDL	CI_EBY_PDSO_SSDL_BAPI

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Customer Orders. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

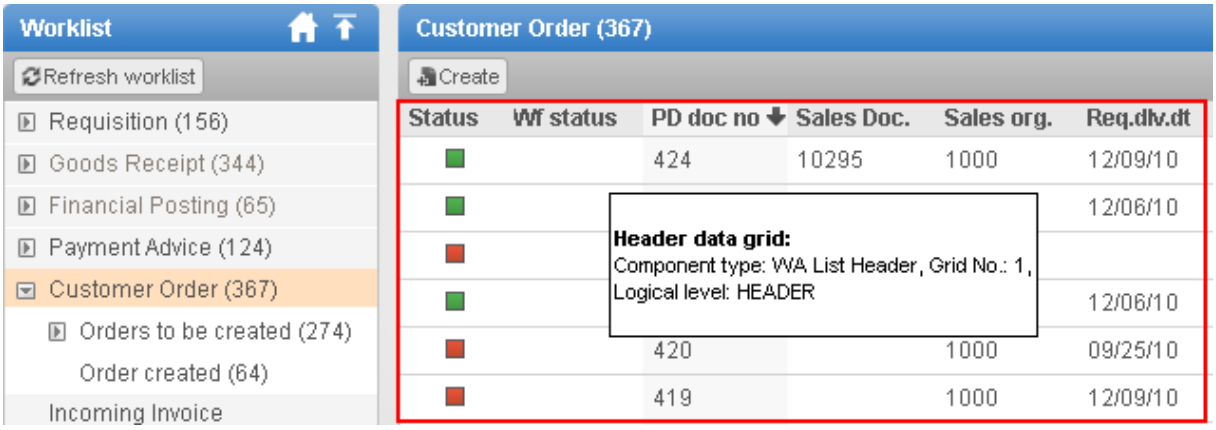
- SAP document overview list:



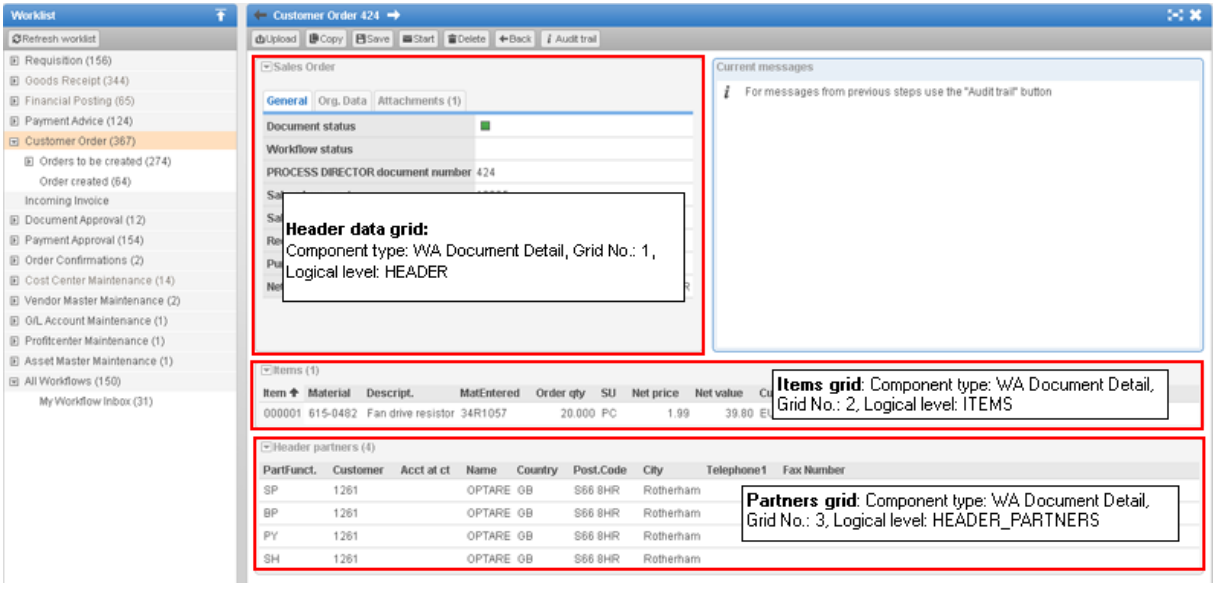
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

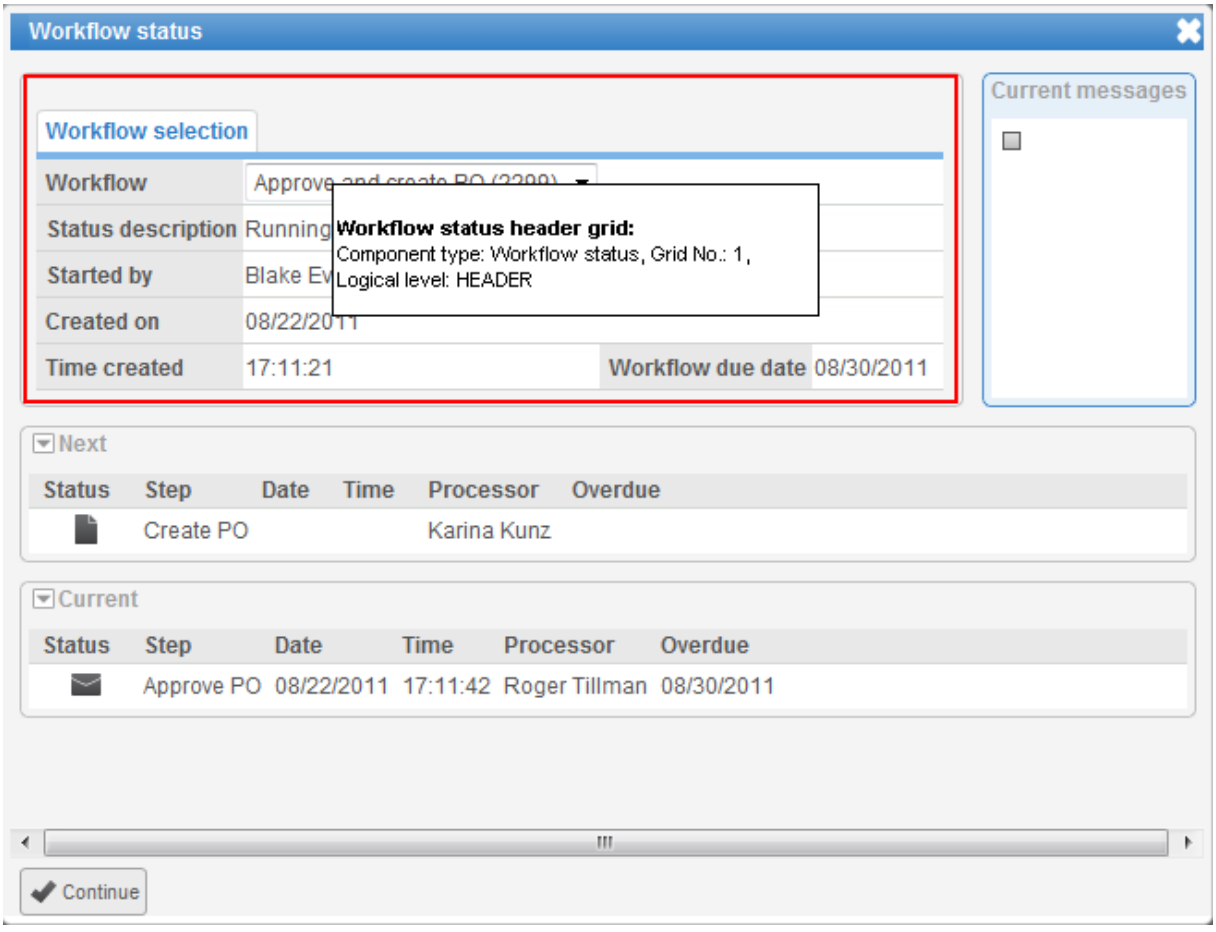
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDSO_SGRID1 ZXY_PDSO_LGRID1	Header Header data
	2	ITEMS	ZXY_PDSO_SGRID2 ZXY_PDSO_LGRID2	Items Items
	3	HEADER_PARTNERS	ZXY_PDSO_SGRID3 ZXY_PDSO_LGRID3	Partners Partners

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
Additional grid 01	1	HEADER_COND	ZXY_PDSO_SGRID4 ZXY_PDSO_LGRID4	Conditions Header conditions
Additional grid 01	2	ITEM_COND	ZXY_PDSO_SGRID6 ZXY_PDSO_SGRID6	Conditions Price conditions
Additional grid 01	3	SCHEDULES	ZXY_PDSO_SGRID5 ZXY_PDSO_LGRID5	Schedules Schedules
WA Document Detail	1	HEADER	ZXY_PDSO_SWC_DOC_GRID1 ZXY_PDSO_LWC_DOC_GRID1	Sales Order Sales Order
WA Document Detail	2	ITEMS	ZXY_PDSO_SWC_DOC_GRID2 ZXY_PDSO_LWC_DOC_GRID2	Items Items
WA Document Detail	3	HEADER_PARTNERS	ZXY_PDSO_SWC_DOC_GRID3 ZXY_PDSO_LWC_DOC_GRID3	Partners Partners
WA List Header	1	HEADER	ZXY_PDSO_SWC_LIST_GRID1 ZXY_PDSO_LWC_LIST_GRID1	Sales Order Sales Order
Workflow status	1	HEADER	ZXY_PDSO_SWC_STAT_GRID1 ZXY_PDSO_LWC_STAT_GRID1	Sales Order Sales Order
Header data for archive log	1	HEADER	ZXY_PDSO_SARC_LOG_GRID1 ZXY_PDSO_LARC_LOG_GRID1	Sales Order Sales Order

Document statuses

PROCESS DIRECTOR ships with these system document statuses for Customer Orders:

Status	Status Description
BG	In background process
DL	Deleted
ER	In error - no sales order
NN	No sales order created

Status	Status Description
OK	Sales order created
XX	Rejected

Text types

PROCESS DIRECTOR ships with these system text types for Customer Orders:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X
HEADER	VBBK	Sales header texts	0002	Header note 1	
HEADER	VBBK	Sales header texts	V002	Sales text	
ITEMS	VBBP	Sales item texts	0001	Material sales text	

User exits / BAdIs

BAdI Posting (Customer Orders)

This user exit / BAdI currently provides the interface methods `BEFORE_POST` and `AFTER_POST` that you can implement to a your own customer logic before or after a Customer Order document is posted.

BEFORE_POST method

Calling sequence

This method is called from the action `POST_DOC Post document`. It runs immediately *before* document posting occurs.

Method signature

Type	Parameter	Description
	IR_OBJECT	Sales order object
	IS_BO_HDR	Data fields of a PD base object
	IS_SO_HDR	Persistent sales order header data
	IT_ITEMS	Customer order items
	IT_HDR_CONDITIONS	Customer order header conditions
	IT_HDR_PARTNERS	Customer order header partners
	IT_ITEM_CONDITIONS	Customer order line item conditions data
	IT_ITEM_PARTNERS	Customer order line item partner data
	IT_ITEM_SCHEDULES	Customer order line item schedules data
	IS_EVENT	Configuration of processes
	CT_CONTRACTS	Table of BAPICTR
	CS_HEADER	Sales order BAPI header
	CT_ITEMS	Sales order BAPI items
	CT_PARTNERS	Sales order BAPI partners
	CT_CONDITIONS	Sales order BAPI conditions
	CT_SCHEDULES	Sales order BAPI schedules
	CT_TEXTS	Sales order BAPI texts

Type	Parameter	Description
	CT_EXTENSIONS	Sales order BAPI extension
	CT_ITEMXS	Table of BAPISDITMX
	CT_SCHEDULESX	Table of BAPISDITMX
	CT_CONDITIONSX	Table of BAPICONDX
	CT_BATCH	Table of BAPIBTSEL
	CT_SCHEDCONF	Table of BAPISCHDL2
	CT_ADRESSES	Table of BAPIADDR1
	CT_CONTRACTSX	Table of BAPICTRX
	CT_KEYS	Table of BAPISDKEY
	CT_CCARD	Table of BAPICCARD
	CT_CONF_REFINST	Table of BAPICUREF
	CT_CONF_VK	Table of BAPICUVK
	CT_CONF_BLOB	Table of BAPICUBLB

Type	Parameter	Description
	CT_CONF_VALUE	Table of BAPICUVAL
	CT_CONF_PARTOF	Table of BAPICUPRT
	CT_CONF_INST	Table of BAPICUINS
	CT_CONF_REF	Table of BAPICUCFG
	CS_HEADERX	Checkbox fields for Sales and Distribution Document Header
	CC_SALESDOCNO	Sales Document
	CS_SENDER	Transfer structure for logical systems
	CC_BINTYPE	Relationship type
	CB_INTNUMBER	
	CC_OBJ	Object type
	CB_CONVERT	
	BC_REFRESHBUFFER	
	CC_WHEN_ERROR	Single-character flag

Type	Parameter	Description
	CC_CALL_ACTIVE	Not more closely defined area, possibly used for patch levels
	CT_CMPB_ASGN	Table of BAPIBTSEL

AFTER_POST method

Calling sequence

This method is called from the action `POST_SUCCESS Posting succeeded`. It runs after the document status `Posted` has been set.

Note: If document posting fails, `AFTER_POST` is *not* called.

Method signature

Type	Parameter	Description
	IS_HEADER	Sales order BAPI header
	IT_ITEMS	Sales order BAPI items
	IT_PARTNERS	Sales order BAPI partners
	IT_CONDITIONS	Sales order BAPI conditions
	IT_SCHEDULES	Sales order BAPI schedules
	IT_TEXTS	Sales order BAPI texts
	IT_EXTENSIONS	Sales order BAPI extension
	CS_BO_HDR	Data fields of a PD base object
	CS_SO_HDR	Persistent sales order header data
	CT_HDR_CONDITIONS	Customer order header conditions

Type	Parameter	Description
	CT_HDR_PARTNERS	Customer order header partners
	CT_ITEMS	Customer order items
	CT_ITEM_CONDITIONS	Customer order item conditions
	CT_ITEM_PARTNERS	Customer order item partners
	CT_ITEM_SCHEDULES	Customer order item schedules

Electronic Bank Statements (ES)

SAP Object type: /EBY/PDES

Package: /EBY/PDES

This chapter provides reference information for Electronic Bank Statements.

Preset classes for Electronic Bank Statements

The following preset class is available for Electronic Bank Statements in the PROCESS DIRECTOR standard. You can also use general preset classes that are available for all process types.

Updates items in the Electronic Bank Statement

/EBY/CL_PDES_PRESET_ITEM

Adds the Payment Advice number and customer number to line items in the Electronic Bank Statement.

Note: Although this preset is a PDES class, it must be added to the Payment Advices presets configuration, not the Electronic Bank Statement presets configuration.

Determinations for Electronic Bank Statements

This determination is available for Electronic Bank Statements in the PROCESS DIRECTOR standard:

ES payment advice determination

`/EBY/CL_PDES_DET_PA_INIT`

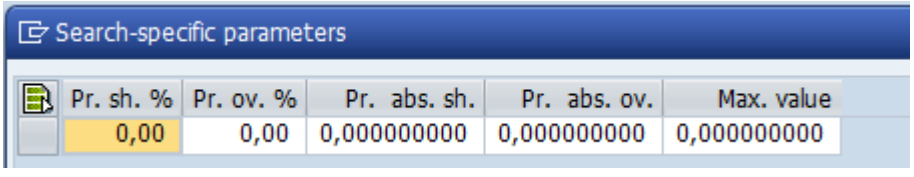
Determines the correct payment advice for an Electronic Bank Statement item by searching for this information in (depending on the defined searches):

Search type	Search name
Search by value	`/EBY/CL_PDES_DET_PA_VALUE`
Search by reference	`/EBY/CL_PDES_DET_PA_REF`

In the initialization parameters, you can define the following rules:

- value shortfall in %
- value overrun in %
- absolute value shortfall
- absolute value overrun
- limit for standard line item value

If one of the defined rules is broken, a message is generated.



Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

Logical level	Package	Interface structure	Customer Includes
Header data	`/EBY/PDES_MODEL`	`/EBY/PDES_SHDR_IF`	`CI_EBY_PDES_SHDR_DATA` `CI_EBY_PDES_SHDR_DISP`
Item data	`/EBY/PDES_MODEL`	`/EBY/PDES_SITM_IF`	`CI_EBY_PDES_SITM_DATA` `CI_EBY_PDES_SITM_DISP`

BAPI structure fields

Some document data models include BAPI structures. These structures can contain a large number of fields, many of which are not used. Therefore, most document data model BAPI structure fields are transient by default.

However, a customer may require that one or more of these fields be made persistent. PROCESS DIRECTOR offers an easy way to fulfill this requirement. In SAP transaction `SE80`, simply add the BAPI structure field(s) to the appropriate customer include.

Logical level	Package	Structure	Customer Include
Header data	`/EBY/PDES_PERSISTENCE`	`/EBY/PDES_THDR`	`CI_EBY_PDES_SHDR_BAPI`
Item data	`/EBY/PDES_PERSISTENCE`	`/EBY/PDES_TITM`	`CI_EBY_PDES_SITM_BAPI`

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Electronic Bank Statements. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

- SAP document overview list:

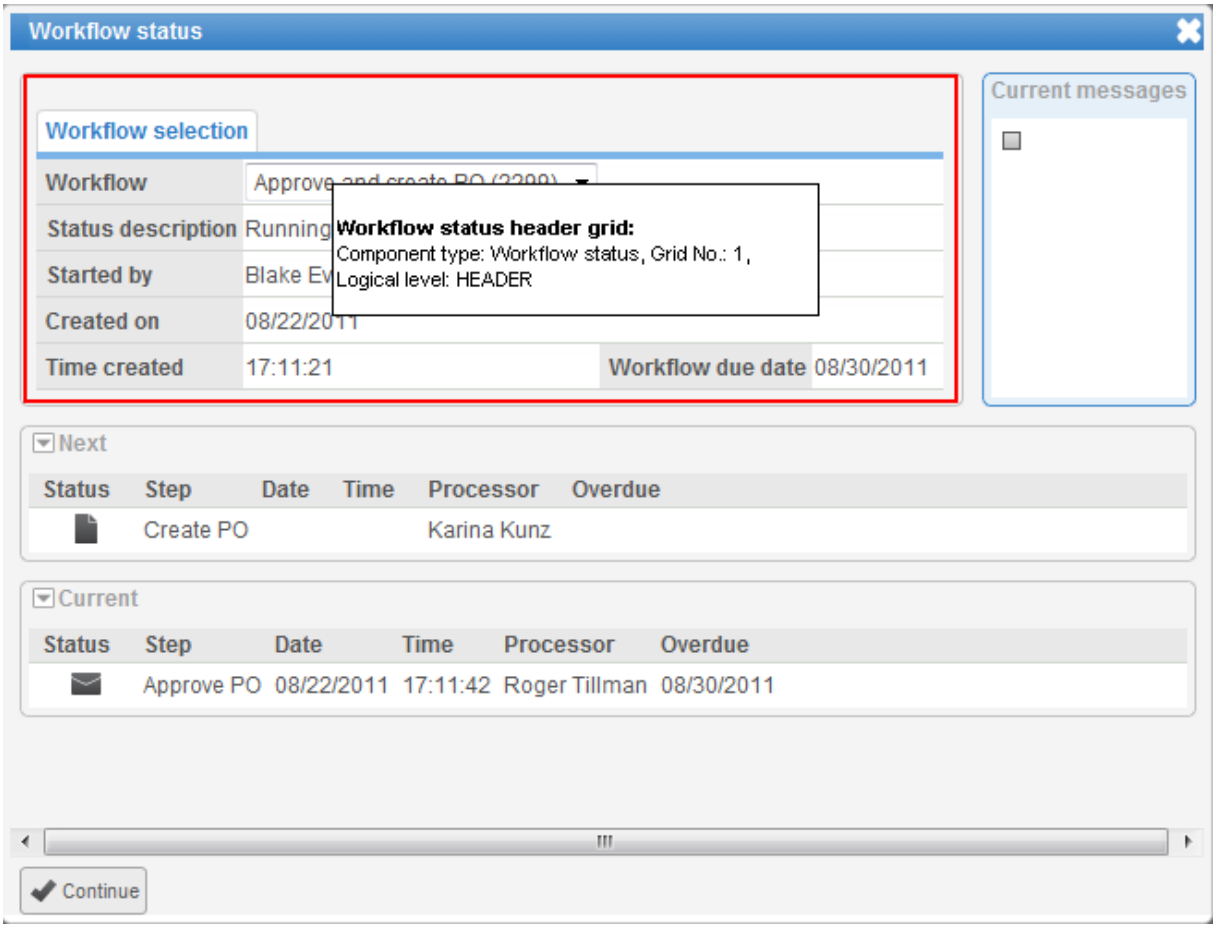
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

- Web Application document overview list:

- Web Application document detail:

- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDES_SGRID1 ZXY_PDES_LGRID1	Header Header data
	2	ITEMS	ZXY_PDES_SGRID2 ZXY_PDES_LGRID2	Items Items
	4	PA_ITEMS	ZXY_PDES_SGRID4 ZXY_PDES_LGRID4	PA Items PA Items
WA Document Detail	1	HEADER	ZXY_PDES_SWC_DOC_GRID1 ZXY_PDES_LWC_DOC_GRID1	Sales Order Sales Order
WA Document Detail	2	ITEMS	ZXY_PDES_SWC_DOC_GRID2 ZXY_PDES_LWC_DOC_GRID2	Items Items
WA Document Detail	4	PA_ITEMS	ZXY_PDES_SWC_DOC_GRID4 ZXY_PDES_LWC_DOC_GRID4	PA Items PA Items
WA List Header	1	HEADER	ZXY_PDES_SWC_LIST_GRID1 ZXY_PDES_LWC_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDES_SWC_STAT_GRID1 ZXY_PDES_LWC_STAT_GRID1	Electronic Bank Statement Electronic Bank Statement
Header data for archive log	1	HEADER	ZXY_PDES_SARC_LOG_GRID1 ZXY_PDES_LARC_LOG_GRID1	Electronic Bank Statement Electronic Bank Statement

Document statuses

PROCESS DIRECTOR ships with these system document statuses for Electronic Bank Statements:

Statuses

Status	Status Description
ER	In error or incomplete

Status	Status Description
NN	New or ready for processing
OK	Processed

Text types

PROCESS DIRECTOR ships with this system text type for Electronic Bank Statements:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

Financial Postings (FI)

SAP Object type: `/EBY/PDFI` (G/L Account Postings), `/EBY/PDFIC` (Customer Postings)

Package: `/EBY/PDFI`

This chapter provides reference information for Financial Postings.

Preset classes for Financial Postings

The following preset classes are available for Financial Postings in the PROCESS DIRECTOR standard. You can also use general preset classes that are available for all process types.

Preset for Credit/Debit Indicator based on posting type

`/EBY/CL_PDFI_PRESET_SHKZG`

Automatically populates the debit/credit indicator on newly added positions.

Note: This preset is only for Customer Postings.

Preset for payment terms

`/EBY/CL_PDFI_PRESETS_PAY_TERMS`

Automatically populates the details of the payment terms. The payment terms are determined from the specified customer and company code.

Note: This preset is only for Customer Postings.

Checks for Financial Postings

The following checks are available for Financial Postings in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check against check tables of the popup fields

`/EBY/CL_PDFI_EVT_CHK_CHECKTABL`

Checks whether the original document values entered in the popup for entering initial data exist and are valid.

This check is pre-configured in the FI document creation process and does not have to be added to the checks configuration.

Checks second popup for Adjustment Postings

`/EBY/CL_PDFI_EVT_CHK_GLAP_DATA`

For Adjustment Postings (process type G/L Account Postings), checks whether source and target cost elements have been entered in the popup for entering initial data.

This check is pre-configured in the FI document creation process and does not have to be added to the checks configuration.

Checks whether data can be posted

`/EBY/CL_PDFI_EVT_CHK_CREATE`

Simulates the creation of the FI document in SAP using the BAPI and returns all SAP messages generated during the simulation.

Checks whether data can be posted via posting interface

`/EBY/CL_PDFI_EVT_CHK_POST_IF`

Simulates the creation of the FI document in SAP using the posting interface and returns all SAP messages generated during the simulation.

Checks whether recurring schedule fields are correct

`/EBY/CL_PDFI_EVT_CHK_SCHEDULE`

For recurring templates, checks whether the entered **Last run date** is later than the entered **First run date**.

Process parameters for Financial Postings

The following process parameters are available for Financial Postings. You can also use the general process parameters that are available for all process types.

Process	Description
Archiving object types to be attached to the provision (Archiving)	
Posting interface parameters (Checks, Posting)	<table><tr><th>Parameter</th><th>Description</th></tr><tr><td>**No authorization check**</td><td></td></tr><tr><td>**Only check**</td><td></td></tr><tr><td>**Update mode**</td><td></td></tr><tr><td>**Display mode of the BDC interface**</td><td></td></tr><tr><td>**Background**</td><td></td></tr></table>
Prepare batch input (Checks, Posting)	
Set posting type for external data reception (Posting)	
Set posting key for write off determination (Posting)	

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

FI G/L Account Postings

Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/PDFI_MODEL	/EBY/PDFI_SHDR_IF	CI_EBY_PDFI_SHDR_DATA CI_EBY_PDFI_SHDR_DISP
Item data	/EBY/PDFI_MODEL	/EBY/PDFI_SITM_IF	CI_EBY_PDFI_SITM_DATA CI_EBY_PDFI_SITM_DISP

FIC Customer Postings

Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/PDFI_MODEL	/EBY/PDFIC_SHDR_IF	CI_EBY_PDFIC_SHDR_DATA CI_EBY_PDFIC_SHDR_DISP
Item data	/EBY/PDFI_MODEL	/EBY/PDFIC_SITM_IF	CI_EBY_PDFIC_SITM_DATA CI_EBY_PDFIC_SITM_DISP

BAPI structure fields

Some document data models include BAPI structures. These structures can contain a large number of fields, many of which are not used. Therefore, most document data model BAPI structure fields are transient by default.

However, a customer may require that one or more of these fields be made persistent. PROCESS DIRECTOR offers an easy way to fulfill this requirement. In SAP transaction `SE80`, simply add the BAPI structure field(s) to the appropriate customer include.

FI G/L Account Postings

Logical level	Package	Structure	Customer Include
Header data	`/EBY/PDFI_PERSISTENCE`	`/EBY/PDFI_THDR`	`CI_EBY_PDFI_SHDR_BAPI`
Item data	`/EBY/PDFI_PERSISTENCE`	`/EBY/PDFI_TITM`	`CI_EBY_PDFI_SITM_BAPI`

FIC Customer Postings

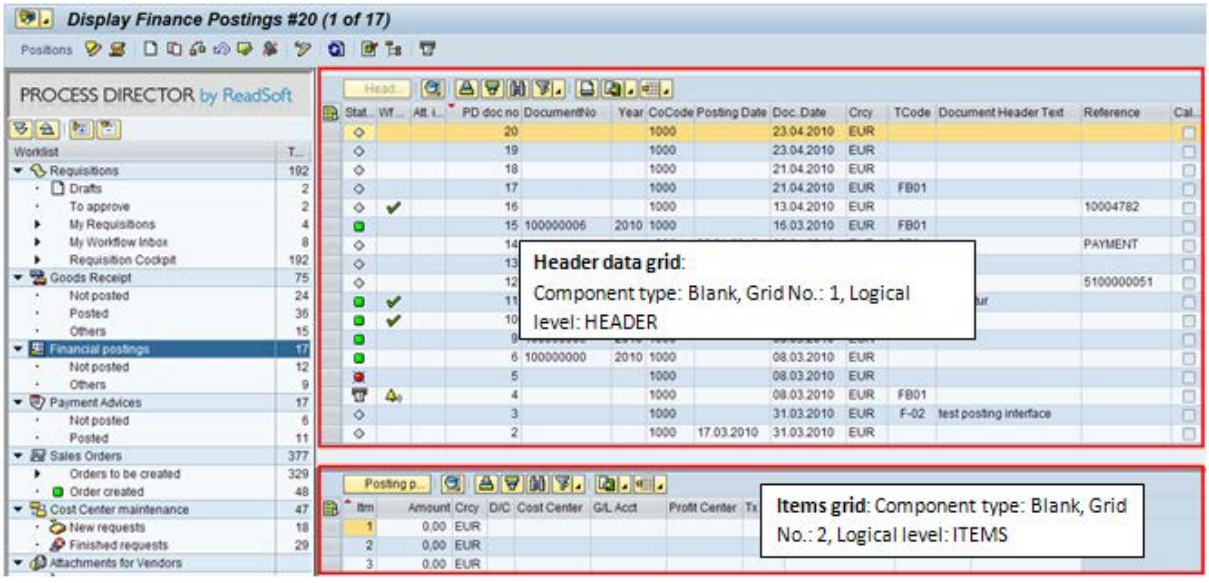
Logical level	Package	Structure	Customer Include
Header data	`/EBY/PDFI_PERSISTENCE`	`/EBY/PDFIC_THDR`	`CI_EBY_PDFIC_SHDR_BAPI`
Item data	`/EBY/PDFI_PERSISTENCE`	`/EBY/PDFIC_TITM`	`CI_EBY_PDFIC_SITM_BAPI`

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Financial Postings. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

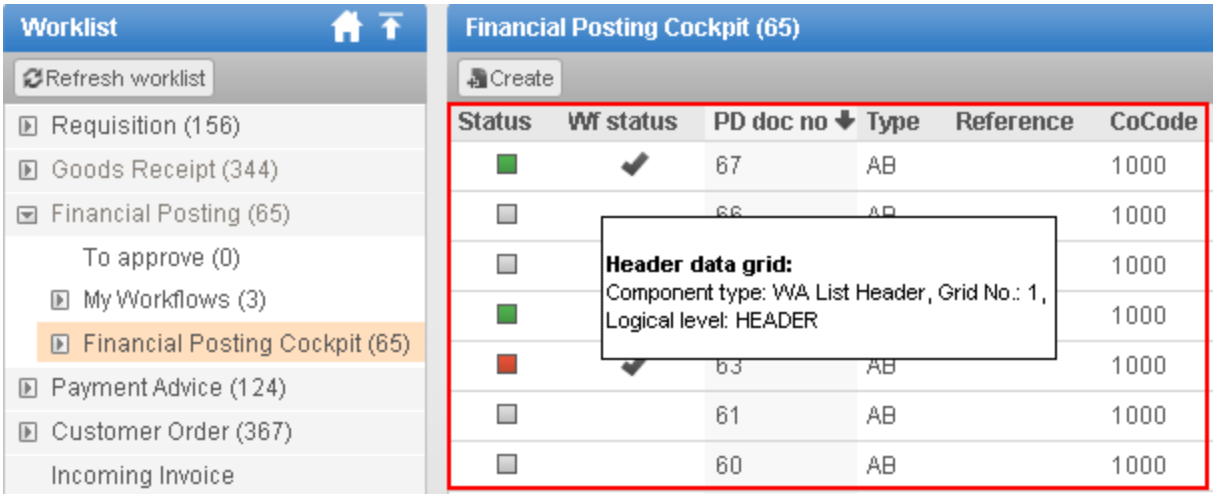
- SAP document overview list:



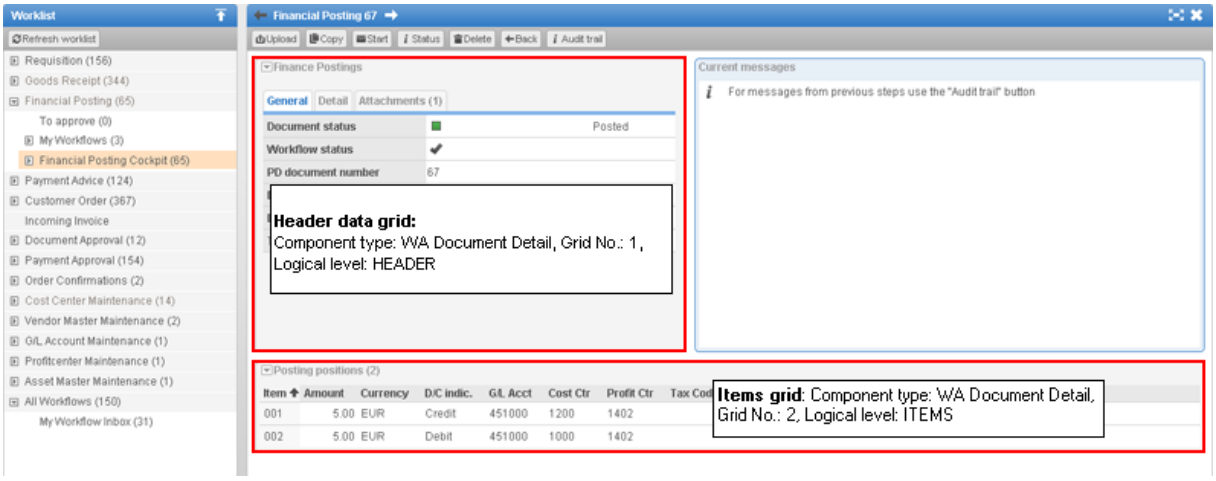
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

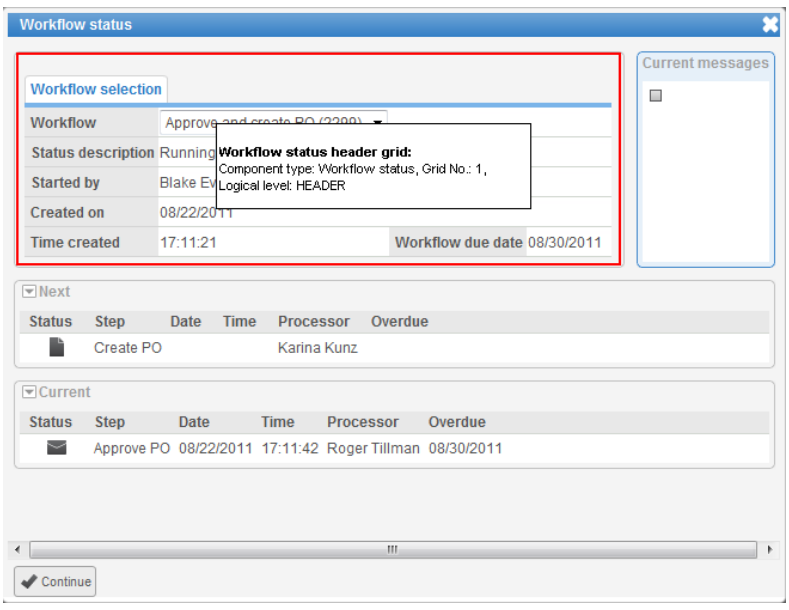
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDFI_SGRID1 ZXY_PDFI_LGRID1	Header Header data
	2	ITEMS	ZXY_PDFI_SGRID2 ZXY_PDFI_LGRID2	Positions Posting positions

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
WA Document Detail	1	HEADER	ZXY_PDFI_SGRID3 ZXY_PDFI_LGRID3	Finance Postings Header
WA Document Detail	2	ITEMS	ZXY_PDFI_SGRID4 ZXY_PDFI_LGRID4	Positions Posting positions
WA List Header	1	HEADER	ZXY_PDFI_SWA_LIST_GRID1 ZXY_PDFI_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDFI_SWA_STAT_GRID1 ZXY_PDFI_LWA_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDFI_SARC_LOG_GRID1 ZXY_PDFI_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses for Financial Postings:

Statuses

Status	Status Description
ER	Error
NN	New or ready for posting
OK	Posted
RE	Reversed
XX	Rejected

Substatuses

FI substatuses are used primarily to assign the appropriate field statuses to FI documents.

Status	Status Description
01	GLSP Regular posting
02	GLPR Provision
03	GLAP Adjustment posting
04	GLRE Recurring entry
05	GLAR Accrual with reversal
06	GLRA Recurring accrual with reversal
07	GLAA Aggregated accrual from PDAP

Text types

PROCESS DIRECTOR ships with this system text type for Financial Postings:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

User exits / BAdIs

BAdI Posting (Financial Postings)

This user exit / BAdI currently provides the interface methods `BEFORE_POST`, `BEFORE_POST_PI`, `AFTER_POST` and `AFTER_POST_PI` that you can implement to add your own customer logic before or after a Financial Posting document is posted, either via the BADI or the posting interface (PI).

BEFORE_POST method

Calling sequence

This method is called from the action `POST_DOC Post document`. It runs immediately *before* document posting occurs.

Method signature

Type	Parameter	Description
	`IR_OBJECT`	Financial postings object
	`IS_BO_HDR`	Data fields of a PD base object
	`IS_FI_HDR`	Financial postings header data
	`IT_ITEMS`	Financal postings item data
	`CS_HEADER`	Header
	`CS_CUSTOMERCPD`	Posting in accounting: Partner billing doc (load receivable)
	`CS_CONTRACTHEADER`	Add. contract Accounts Receivable and Accounts Payable header line
	`CT_ACCOUNTGL`	G/L account item
	`CT_ACCOUNTRECEIVABLE`	Customer item
	`CT_ACCOUNTPAYABLE`	Vendor item
	`CT_ACCOUNTTAX`	Tax item
	`CT_CURRENCYAMOUNT`	Currency items
	`CT_CRITERIA`	CO-PA account assignment characteristics

Type	Parameter	Description
	CT_VALUEFIELD	CO-PA account assignment value fields
	CT_EXTENSION1	Container for 'Customer Exit' parameter
	CT_EXTENSION2	BAPI Extension

BEFORE_POST_PI method

Method signature

Type	Parameter	Description
	IR_OBJECT	Financial postings object
	IS_BO_HDR	Data fields of a PD base object
	IS_FI_HDR	Financial postings header data
	IT_ITEMS	Financal postings item data
	CC_TCODE	Transaction code
	CB_SGFUNCT	FI posting interface alternative function
	CT_BLNTAB	Document number table for financial accounting
	CT_FTPOST	Document header and items for internal posting interface
	CT_FTTAX	Taxes for internal posting interface
	CT_RETURN	Return parameter table
	CB_NO_AUTH	Deactivate authorization check

Type	Parameter	Description
	CB_CHECK	Only check
	CC_UPDATEMODE	Update mode for call transaction
	CC_CTU_MODE	Batch input processing mode
	CB_DARK	Transactions to be processed in the background

AFTER_POST method

Calling sequence

In the PROCESS DIRECTOR standard, this user exit is called from the action `POST_SUCCESS` `Posting succeeded`. It runs after the document status has been set.

Note: If document posting fails, `AFTER_POST` is *not* called.

Method signature

Type	Parameter	Description
	IR_OBJECT	Financial postings object
	IS_HEADER	Header
	IS_CUSTOMERCPD	Posting in accounting: Partner billing doc (load receivable)
	IT_CONTRACTHEADER	Add. contract Accounts Receivable and Payable Header Line
	IT_ACCOUNTGL	G/L account item
	IT_ACCOUNTRECEIVABLE	Customer item
	IT_ACCOUNTPAYABLE	Vendor item

Type	Parameter	Description
	IT_ACCOUNTTAX	Tax item
	IT_CURRENCYAMOUNT	Currency Items
	IT_CRITERIA	CO-PA account assignment characteristics
	IT_VALUEFIELD	CO-PA account assignment value fields
	IT_EXTENSION1	Container for 'Customer Exit' Parameter
	CS_BO_HDR	Data fields of a PD base object
	CS_FI_HDR	Financial postings header data
	CT_ITEMS	Financal postings item data

AFTER_POST_PI method

Method signature

Type	Parameter	Description
	IR_OBJECT	Financial postings object
	IC_TCODE	Transaction code
	IB_SGFUNCT	FI posting interface alternative function
	IT_BLNTAB	Document number table for financial accounting
	IT_FTPOST	Document header and items for internal posting interface
	IT_FTTAX	Taxes for internal posting interface

Type	Parameter	Description
	IT_RETURN	Return parameter table
	IB_NO_AUTH	Deactivate authorization check
	IB_CHECK	Only check
	IC_UPDATEMODE	Update mode for call transaction
	IC_CTU_MODE	Batch input processing mode
	IB_DARK	Transactions to be processed in the background
	CS_BO_HDR	Data fields of a PD base object
	CS_FI_HDR	Financial postings header data
	CT_ITEMS	Financal postings item data

Programs

The following programs are available for use with Financial Postings:

- /EBY/PDFI_GEN_RECURRING_DOCS

  Automatically generates FI documents from recurring templates.

- /EBY/PDFI_PROVISION_REMINDER

  Sends an email to the person who created the provision document when the validity date of the posted document is exceeded.

/EBY/PDFI_GEN_RECURRING_DOCS

Use the /EBY/PDFI_GEN_RECURRING_DOCS program to automatically generate PROCESS DIRECTOR Financial Posting documents from recurring templates.

Setting	Description
Company code	Company codes for which documents should be created.
Posting type	PROCESS DIRECTOR posting types for which documents should be created.
Only test general sel. options	Check this option if you do not want to create any documents, but only need information about the number of documents that would be created if the program was run.
Current processor	The name specified here will be inserted as the creator of the generated documents.
PROCESS DIRECTOR user type	Enter the user type of the specified current processor: Internet, LDAP or SAP.

/EBY/PDFI_PROVISION_REMINDER

Use the `/EBY/PDFI_PROVISION_REMINDER` program to send an email to the person who created the provision document when the validity date of the posted document is exceeded. PROCESS DIRECTOR provides a standard email text for this purpose.

Setting	Description
Company code	Company codes for which provision emails should be created.
Posting type	Select the posting type `GLPR`.
Only test general sel. options	Check this option if you do not want to send any emails, but only need information about the number of documents for which emails would be sent if the program was run.
Current processor	The name specified here will be inserted as the creator of the generated documents.
PROCESS DIRECTOR user type	Enter the user type of the specified current processor: Internet, LDAP or SAP.

Generic archiving (ARCH_BUS)

SAP Object type: /EBY/PDXY

Package: /EBY/PDAB

This chapter provides reference information for Generic Archiving.

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

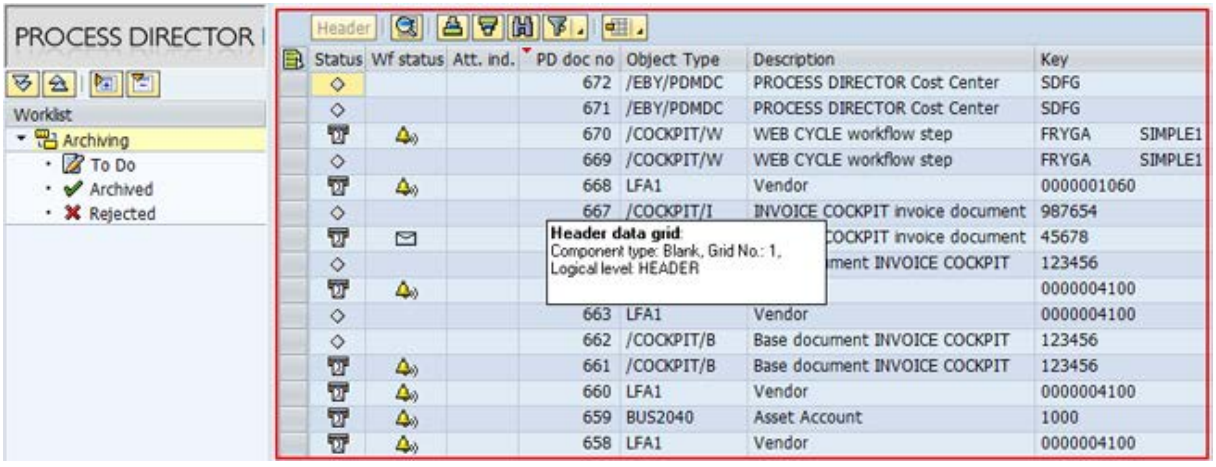
Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/PDAB_MODEL	/EBY/PDAB_SHDR_IF	CI_EBY_PDAB_SHDR_DATA CI_EBY_PDAB_SHDR_DISP

View model customization

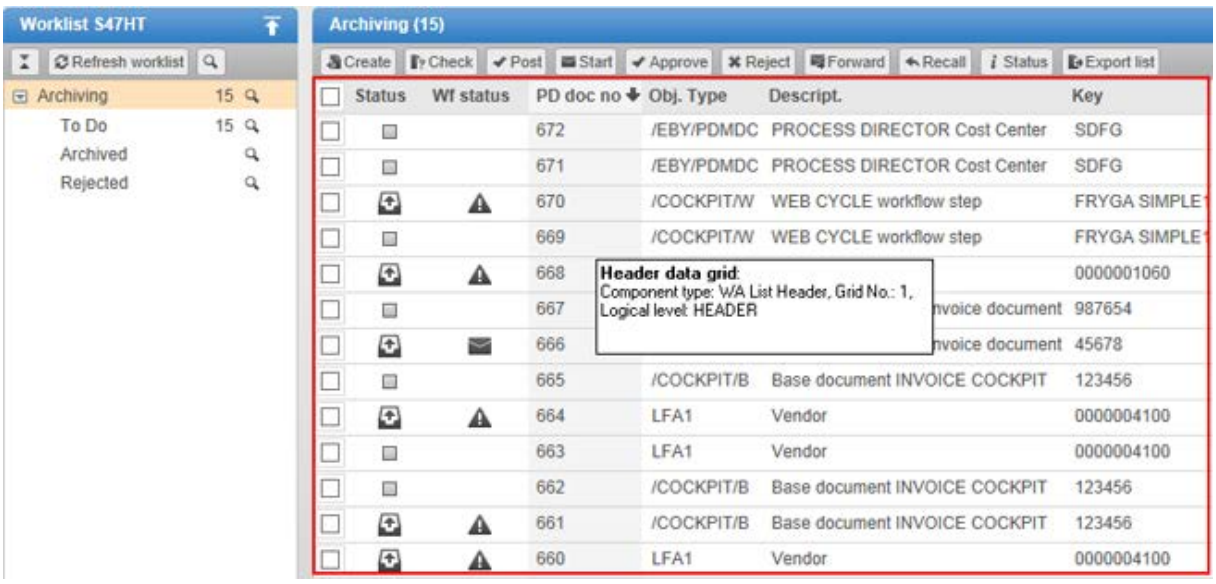
This section provides detailed information about which settings to enter when creating a customer view model for Generic Archiving. See the *PROCESS DIRECTOR Configuration Guide* for more information.

By GUI element

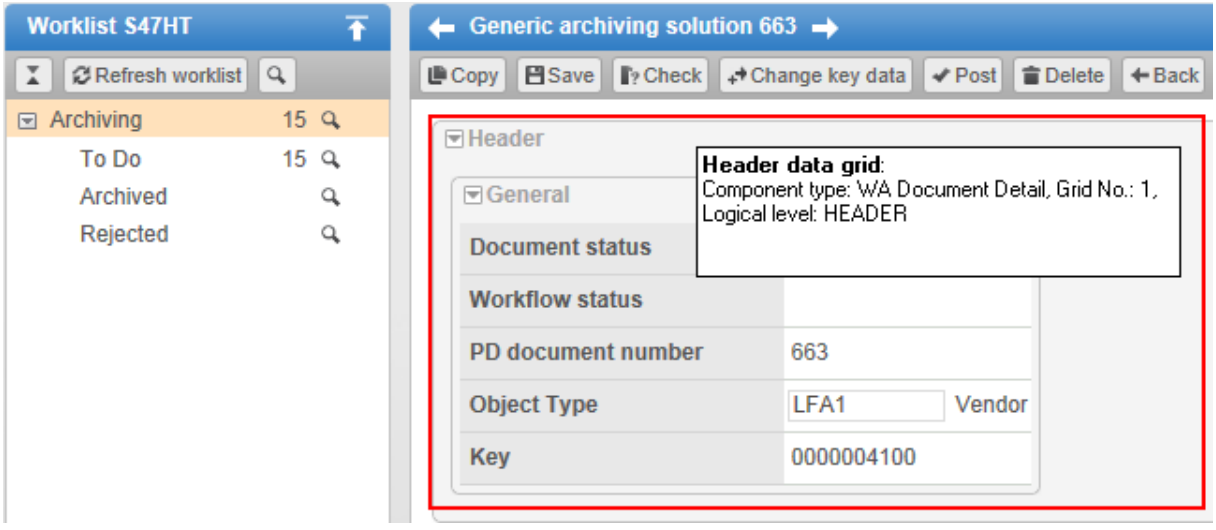
- SAP document overview list:



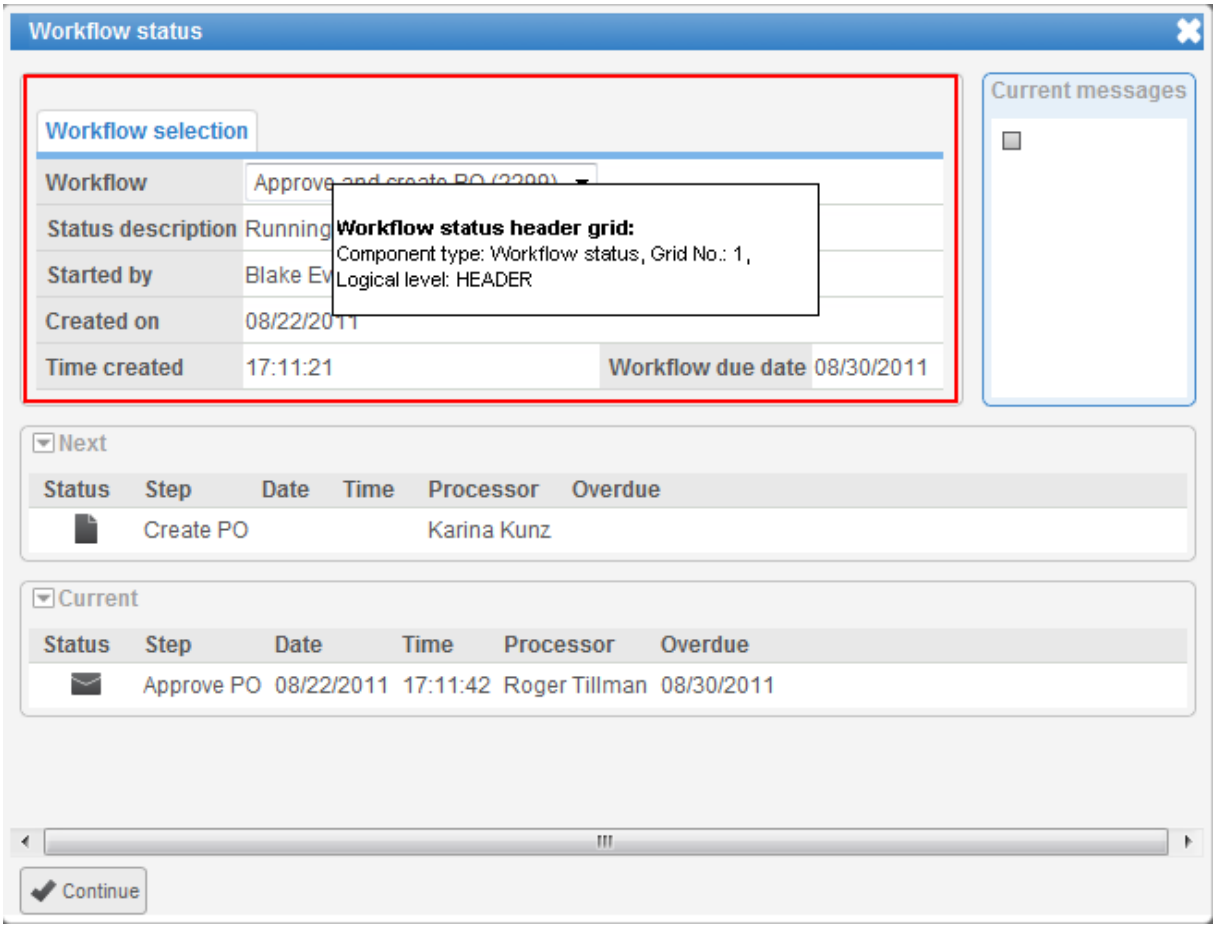
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDAB_SGRID1 ZXY_PDAB_LGRID1	Header Header data
WA Document Detail	1	HEADER	ZXY_PDAB_SWA_DOC_GRID1 ZXY_PDAB_LWA_DOC_GRID1	Header Header data
WA List Header	1	HEADER	ZXY_PDAB_SWA_LIST_GRID1 ZXY_PDAB_LWA_LIST_GRID1	Header Header
Workflow status	1	HEADER	ZXY_PDAB_SWA_STAT_GRID1 ZXY_PDAB_LWA_STAT_GRID1	Header Header
Header data for archive log	1	HEADER	ZXY_PDAB_SARC_LOG_GRID1 ZXY_PDAB_LARC_LOG_GRID1	Header Header

Document statuses

PROCESS DIRECTOR ships with these system document statuses/substatuses for Generic Archiving:

Statuses

Status	Status Description
ER	In error or incomplete
NN	New or ready for processing
OK	Processed
XX	Rejected

Text types

PROCESS DIRECTOR ships with this system text type for Generic Archiving:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

Goods Receipts (DN)

SAP Object type: /EBY/PDDN

Package: /EBY/PDDN

This chapter provides reference information for Goods Receipts.

Preset classes for Goods Receipts

The following preset classes are available for Goods Receipts in the PROCESS DIRECTOR standard. You can also use general preset classes that are available for all process types.

Copy the PO from header to item level

/EBY/CL_PDDN_PRESET_PO2ITM

Copies the purchasing document number from the header to the line items.

Preset DN header from user settings

/EBY/CL_PDDN_PRESET_HEADER

Presets values in the goods receipt document using the values contained in the user profile.

Preset DN vendor from PO or name from LIFNR

/EBY/CL_PDDN_PRESET_VENDOR

Presets the vendor number using the value given in the purchase order, or presets the vendor name using the value from the **Vendor** field.

Checks for Goods Receipts

The following checks are available for goods receipts in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check whether a goods movement can be posted

`/EBY/CL_PDDN_EVT_CHK_CREATE`

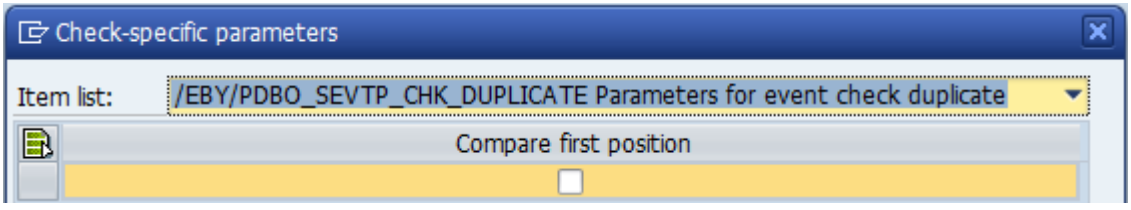
Simulates the creation of the goods movement in SAP and returns all SAP messages generated during the simulation.

Check whether goods movement has already been posted

`/EBY/CL_PDDN_EVT_CHK_DUPLICATE`

Checks whether the same customer has already ordered with the given PO number.

In the initialization parameters, you can specify that the PO number and also the first line item of the PO should be used (item number, material and quantity are compared).



Check vendor of added PO items

`/EBY/CL_PDDN_EVT_CHK_VEND_ITM`

Checks whether new goods receipt line items added using the **Add PO items** function have the same vendor as existing line items.

Determinations for Goods Receipts

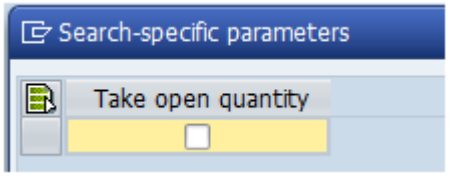
These determinations are available for Goods Receipts in the PROCESS DIRECTOR standard:

Delivery Note item proposal as determination

`/EBY/CL_PDDN_DET_ITEMPROP_INIT`

Adds line items from the purchase order to the goods receipt.

If only open items (that is, items for which the full quantity has not yet been delivered) should be proposed, check the **Take open quantity** checkbox in the search parameters of the **Delivery Notes item proposal from PO** search (`/EBY/CL_PDDN_DET_ITEMPROP`). Only the open quantity is transferred to the goods receipt line items.



Delivery Notes Material determination

`/EBY/CL_PDDN_DET_MAT_INIT`

Determines the material number by checking whether the material number given on the delivery note is (depending on the defined searches):

Search area	Search name
The material number from the material master data	`/EBY/CL_PDDN_DET_MAT_MARA`
A manufacturer part number maintained in the material master data	`/EBY/CL_PDDN_DET_MAT_MANU`
An EAN maintained in the material master data	`/EBY/CL_PDDN_DET_MAT_EAN`
A vendor material number maintained in the table `EINA`	`/EBY/CL_PDDN_DET_MAT_EINA`
The vendor material number given on the purchase order	`/EBY/CL_PDDN_DET_MAT_VENDOR`

Delivery Note Plant determination

`/EBY/CL_PDDN_DET_PLANT_INIT`

Determines the correct plant and PO unit for a goods receipt line item by searching for this information in (depending on the defined searches):

Search area	Search name
The material master data	`/EBY/CL_PDDN_DET_PLANT_MAT`
The purchase order	`/EBY/CL_PDDN_DET_PLANT_PO`

Delivery Notes Purchase Order determination

`/EBY/CL_PDDN_DET_PO_INIT`

Verifies the purchase order number given on the delivery note by searching the `EKKO` (purchasing document header) table for this number.

Delivery Note purchase order line item determination

`/EBY/CL_PDDN_DET_POPOS_INIT`

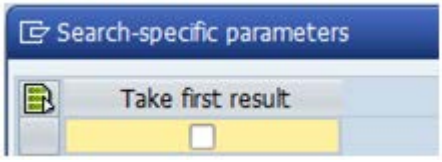
Determines the purchase order line item number by comparing the material and delivery quantity given on the delivery note with the material and undelivered order quantity on the purchase order.

An exact match takes preference over a partial delivery.

The PO line item number can be determined for (depending on the defined searches):

Search area	Search name
The material	`/EBY/CL_PDDN_DET_POPOS_MAT`
The customer material	`/EBY/CL_PDDN_DET_POPOS_IDNLF`
The quantity	`/EBY/CL_PDDN_DET_POPOS_QTY`

In the search parameters, you can specify that the system should use the first match found.



Delivery Note vendor determination

`/EBY/CL_PDDN_DET_VENDOR_INIT`

Verifies the vendor based on (depending on the defined searches):

Search area	Search name
The account number given on the delivery note	`/EBY/CL_PDDN_DET_VENDOR_ACC`

Search area	Search name
The account number given on the purchase order	`/EBY/CL_PDDN_DET_VENDOR_ACCPO`
The vendor number given on the delivery note	`/EBY/CL_PDDN_DET_VENDOR_OK`
The vendor number given on the purchase order	`/EBY/CL_PDDN_DET_VENDOR_PO`

Process parameters for Goods Receipts

The following process parameters are available for Goods Receipts. You can also use the general process parameters that are available for all process types.

Process	Description
Save substate of document (Checks)	Turn on Field name Too early Too late Early/late
Posting parameters (Posting)	Automatic serial numbers

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

Logical level	Package	Interface structure	Customer Includes
Header data	`/EBY/PDDN_MODEL`	`/EBY/PDDN_SHDR_IF`	`CI_EBY_PDDN_SHDR_DATA` `CI_EBY_PDFI_SHDR_DISP`
Item data	`/EBY/PDDN_MODEL`	`/EBY/PDDN_SITM_IF`	`CI_EBY_PDDN_SITM_DATA` `CI_EBY_PDDN_SITM_DISP`
PO line items	`/EBY/PDDN_MODEL`	`/EBY/PDDN_SEKPO_IF`	`CI_EBY_PDDN_SEKPO_DATA` `CI_EBY_PDDN_SEKPO_DISP`
Item serial numbers	`/EBY/PDDN_MODEL`	`/EBY/PDDN_SSER_IF`	`CI_EBY_PDDN_SSER_DATA` `CI_EBY_PDDN_SSER_DISP`

BAPI structure fields

Some document data models include BAPI structures. These structures can contain a large number of fields, many of which are not used. Therefore, most document data model BAPI structure fields are transient by default.

However, a customer may require that one or more of these fields be made persistent. PROCESS DIRECTOR offers an easy way to fulfill this requirement. In SAP transaction `SE80`, simply add the BAPI structure field(s) to the appropriate customer include.

Logical level	Package	Structure	Customer Include
Header data	`/EBY/PDDN_PERSISTENCE`	`/EBY/PDDN_THDR`	`CI_EBY_PDDN_SHDR_BAPI`
Item data	`/EBY/PDDN_PERSISTENCE`	`/EBY/PDDN_TITM`	`CI_EBY_PDDN_SITM_BAPI`

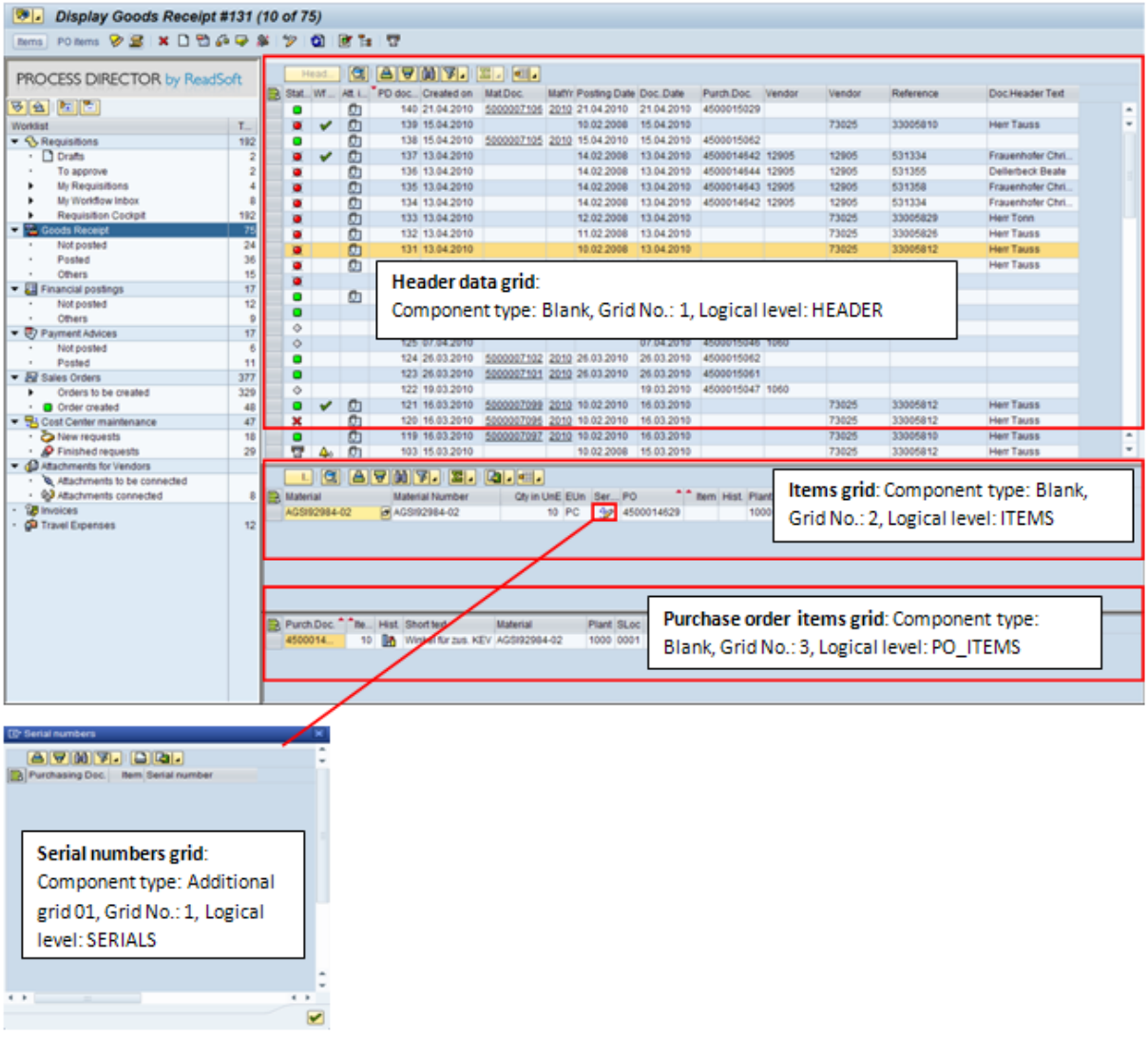
Logical level	Package	Structure	Customer Include
Item serial number data	/EBY/PDDN_PERSISTENCE	/EBY/PDDN_TSER	CI_EBY_PDDN_SSER_BAPI

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Goods Receipts. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

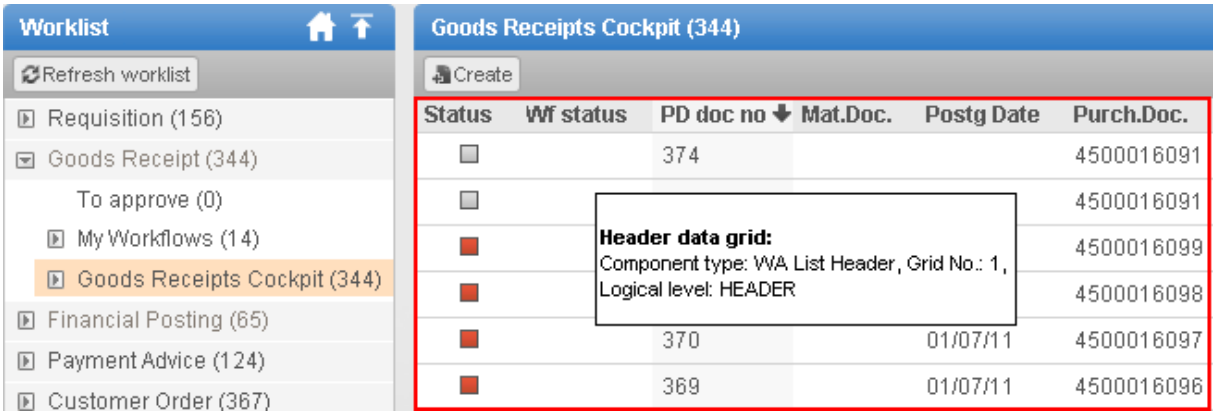
- SAP document overview list:



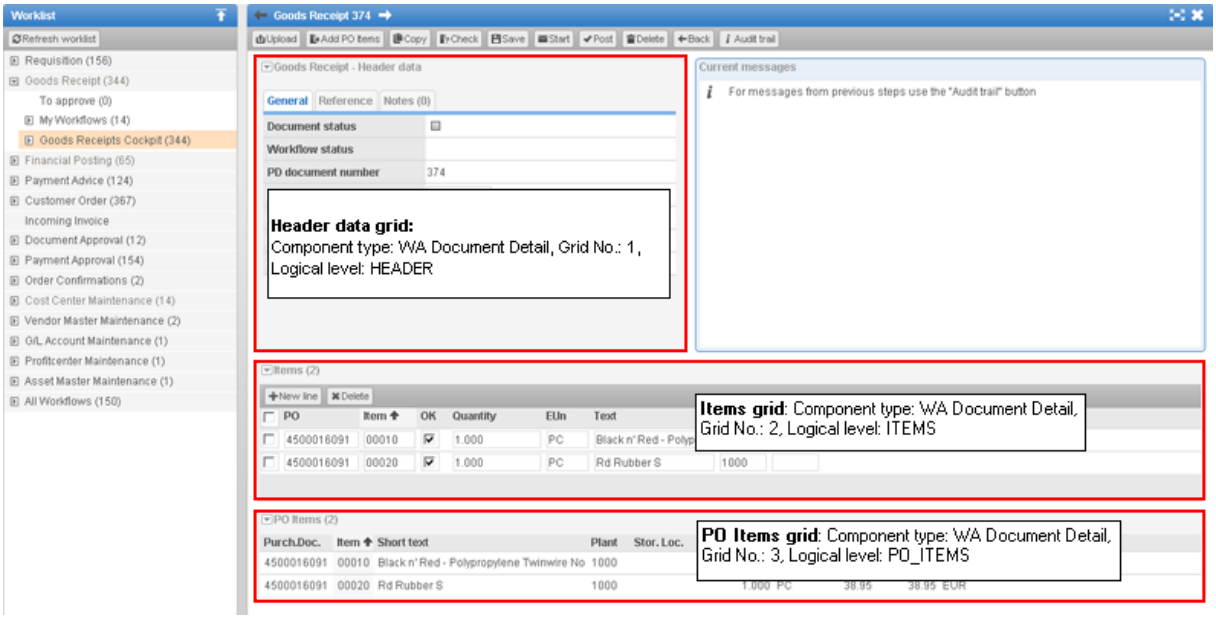
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

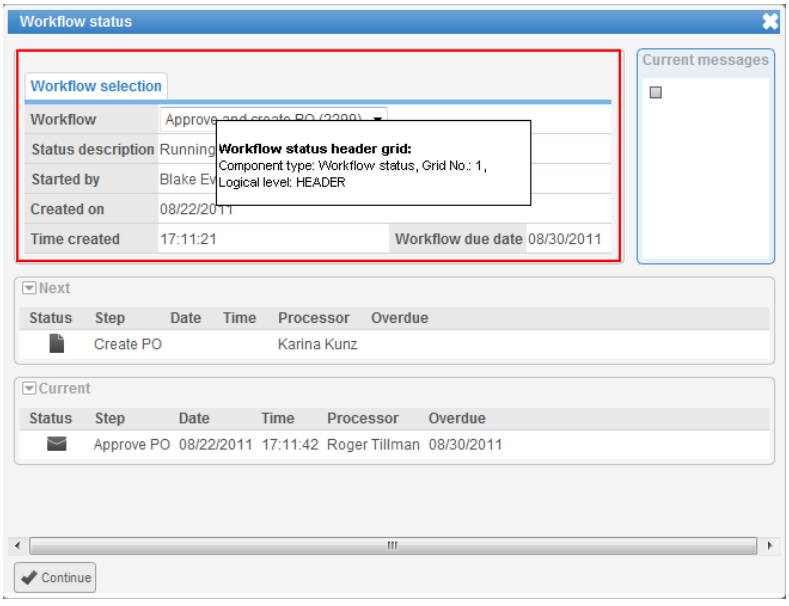
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDDN_SGRID1 ZXY_PDDN_LGRID1	Header Header data
	2	ITEMS	ZXY_PDDN_SGRID2 ZXY_PDDN_LGRID2	Items Items
	3	PO_ITEMS	ZXY_PDDN_SGRID3 ZXY_PDDN_LGRID3	PO items PO items
Additional grid 01	1	SERIALS	ZXY_PDDN_SGRID4 ZXY_PDDN_LGRID4	Serial numbers Serial numbers
WA Document Detail	1	HEADER	ZXY_PDDN_SWA_DOC_GRID1 ZXY_PDDN_LWA_DOC_GRID1	Header Header data
WA Document Detail	2	ITEMS	ZXY_PDDN_SWA_DOC_GRID2 ZXY_PDDN_LWA_DOC_GRID2	Items Items
WA Document Detail	3	PO_ITEMS	ZXY_PDDN_SWA_DOC_GRID3 ZXY_PDDN_LWA_DOC_GRID3	PO items PO items
WA List Header	1	HEADER	ZXY_PDDN_SWA_LIST_GRID1 ZXY_PDDN_LWA_LIST_GRID1	Header Header

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
Workflow status	1	HEADER	ZXY_PDDN_SWA_STAT_GRID1 ZXY_PDDN_LWA_STAT_GRID1	Header Header
Header data for archive log	1	HEADER	ZXY_PDDN_SARC_LOG_GRID1 ZXY_PDDN_LARC_LOG_GRID1	Header Header

Document statuses

PROCESS DIRECTOR ships with these system document statuses for Goods Receipts:

Status	Status Description
CA	Goods movement cancelled
DL	Deleted
ER	In error - no goods movement
NN	No goods movement posted
OK	Goods movement posted
XX	Rejected

Text types

PROCESS DIRECTOR ships with this system text type for Goods Receipts:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

User exits / BAdIs

BAdI Posting (Goods Receipts)

This user exit / BAdI currently provides the interface methods `BEFORE_POST`, `BEFORE_POST_BI`, `AFTER_POST` and `AFTER_POST_BI` that you can implement to add your own customer logic before or after a Goods Receipt document is posted.

BEFORE_POST method

Calling sequence

This method is called from the action `POST_DOC Post document`. It runs immediately *before* document posting occurs.

Method signature

Type	Parameter	Description
	`VALUE(IR_OBJECT)`	Delivery notes object
	`IS_BO_HDR`	Data fields of a PD base object
	`IS_DN_HDR`	Delivery notes header data
	`IT_ITEMS`	Delivery notes DN items
	`IT_PO_ITEMS`	Generic PO items data
	`IT_SERIALS`	Delivery notes serial numbers
	`CS_HEADER`	BAPI Communication Structure: Material Document Header Data
	`CS_CODE`	MMIM: New Key Assignment GM_CODE to Transaction of INV. Mgmt
	`CT_ITEMS`	BAPI Table Type: Create Material Document Item

Type	Parameter	Description
	CT_SERIALS	BAPI Communication Structure: Create Mat. Doc., Serial No.
	CB_XNAPR	Print via output control

BEFORE_POST_BI method

Method signature

Type	Parameter	Description
	VALUE(IR_OBJECT)	Delivery notes object
	IS_BO_HDR	Data fields of a PD base object
	IS_DN_HDR	Delivery notes header data
	IT_ITEMS	Delivery notes DN items
	IT_PO_ITEMS	Generic PO items data
	IT_SERIALS	Delivery notes serial numbers
	CT_BDCDATA	Batch Input Branch to Objects
	CC_TCODE	Transaction code

AFTER_POST method

Calling sequence

This method is called from the action `POST_SUCCESS Posting succeeded`. It runs after the document status `Posted` has been set.

Note: If document posting fails, `AFTER_POST` is *not* called.

Method signature

Type	Parameter	Description
	VALUE(IR_OBJECT)	Delivery notes object
	IS_HEADER	BAPI Communication Structure: Material Document Header Data
	IS_CODE	MMIM: New Key Assignment GM_CODE to Transaction of Inv. Mgmt
	IT_ITEMS	BAPI Table Type: Create Material Document Item
	IT_SERIALS	BAPI Communication Structure: Create Mat. Doc., Serial No.
	IB_XNAPR	Print via output control
	IT_PO_ITEMS	Generic PO items data
	CS_BO_HDR	Data fields of a PD base object
	CS_DN_HDR	Delivery notes header data
	CT_ITEMS	Delivery notes DN items
	CT_SERIALS	Delivery notes serial numbers

AFTER_POST_BI method

Method signature

Type	Parameter	Description
	VALUE(IR_OBJECT)	Delivery notes object

Type	Parameter	Description
	IT_PO_ITEMS	Generic PO items data
	IT_BDCDATA	Batch Input Branch to Objects
	IC_TCODE	Transaction code
	IT_MESSAGES	Collecting messages in the SAP system
	IS_OPTIONS	Parameter string for runtime of CALL TRANSACTION USING ...
	II_RC	Return code after CALL TRANSACTION
	CS_BO_HDR	Data fields of a PD base object
	CS_DN_HDR	Delivery notes header data
	CT_ITEMS	Delivery notes DN items
	CT_SERIALS	Delivery notes serial numbers

Invoice Block and Cancelation (IB)

SAP Object type: `/EBY/PDIB`

Package: `/EBY/PDIB`

This chapter provides reference information for Invoice Block/Cancelation.

Checks for Invoice Block/Cancelation

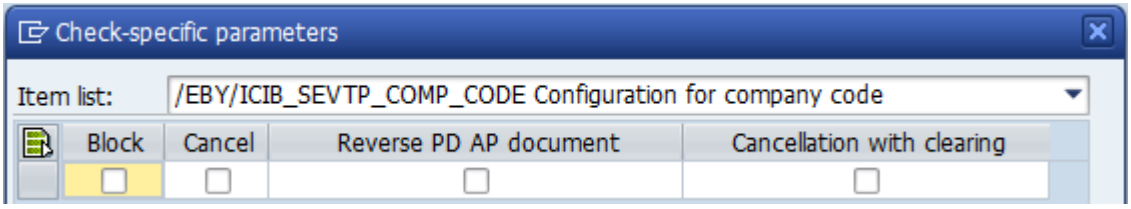
The following checks are available for Invoice Block/Cancelation in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check allowed transaction type for company code

`/EBY/CL_ICIB_EVT_CHK_COMP_CODE`

Checks whether specific actions are allowed for the company code given in the invoice.

Specify which actions should be possible for the company code in the initialization parameters.

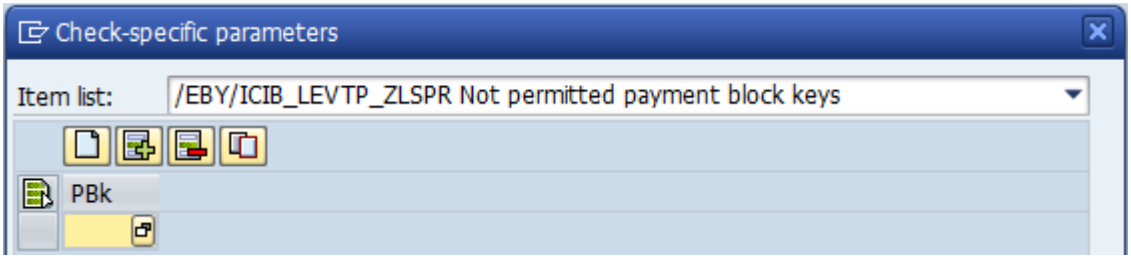


Initialization parameter	Description
Block	Allows invoice blocking.
Cancel	Allows invoice cancelation.
Reverse PD AP document	Allows reversal of the PROCESS DIRECTOR Accounts Payable document.
Cancelation with clearing	Allows cancelation with clearing.

Check for a valid payment block key

Checks whether an entered payment block can be used.

Specify which payment blocks are *not* allowed in the initialization parameters.



Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure,

you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

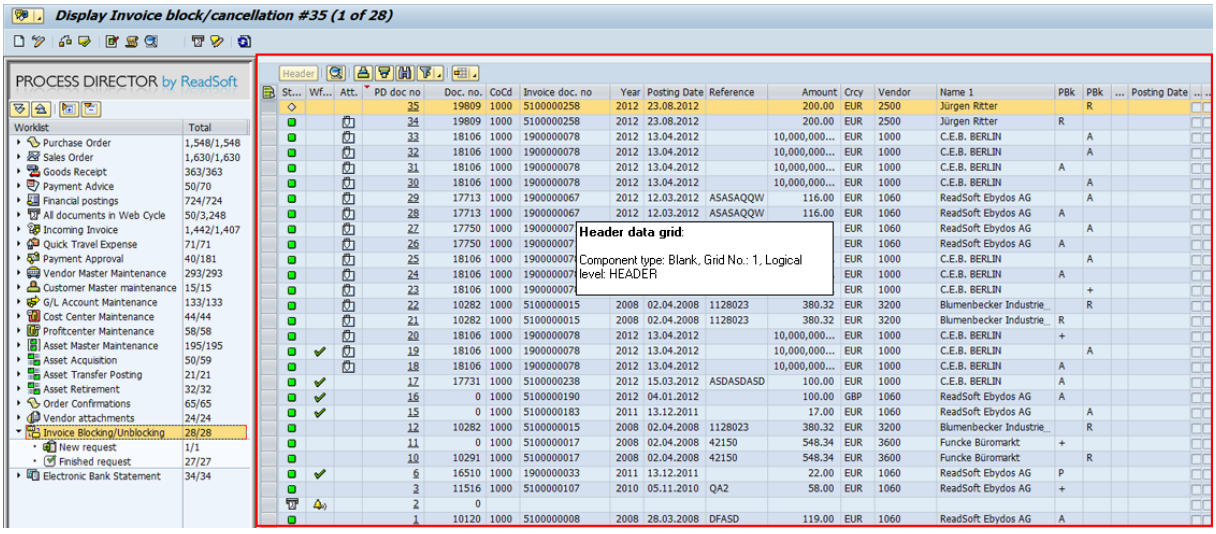
Logical level	Package	Interface structure	Customer Includes
Header data	`/EBY/ICIB_MODEL`	`/EBY/ICIB_SHDR_IF`	`CI_EBY_ICIB_SHDR_DATA`

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Invoice Block/Cancelation. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

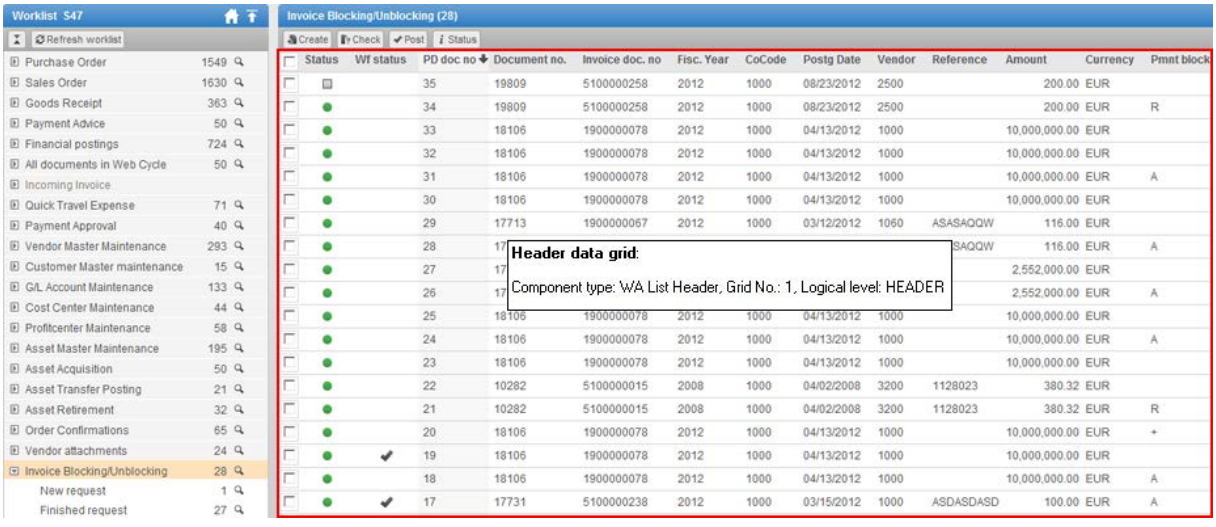
- SAP document overview list:



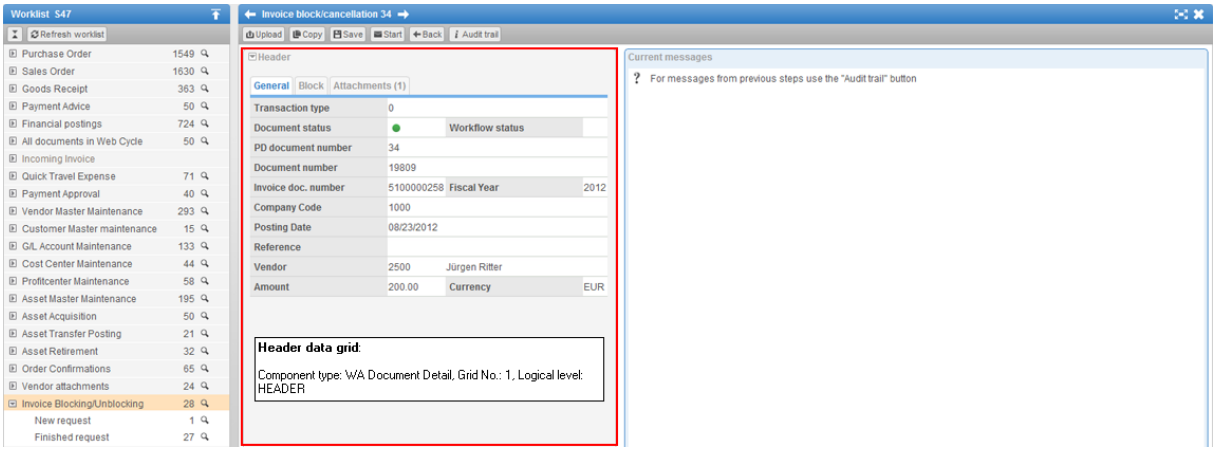
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

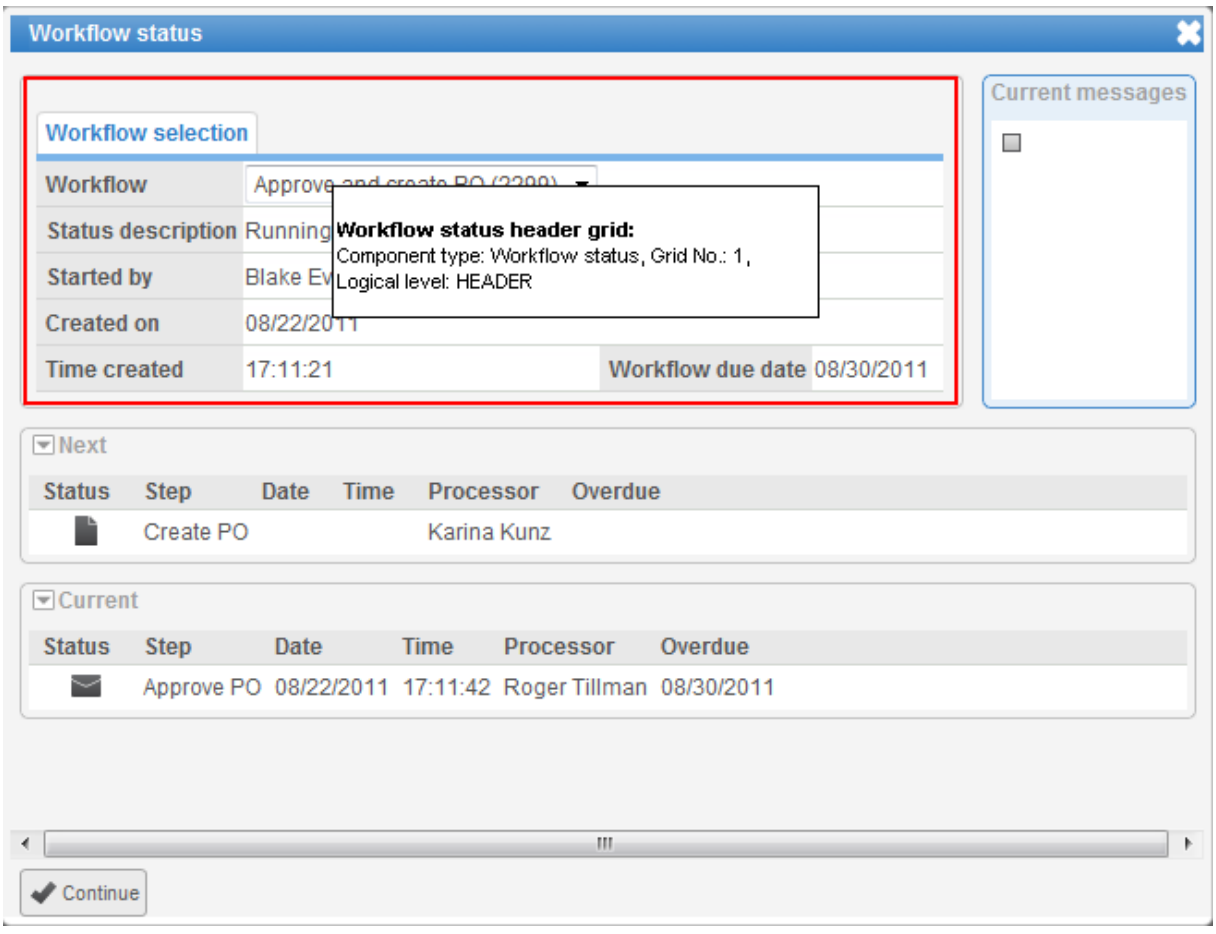
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

GUI area	Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming suggestions)	Btn. label Grid title (naming suggestions)
SAP document list		1	HEADER	ZXY_PDIB_SGRID1 ZXY_PDIB_LGRID1	Header Header data
Web Application document detail header data	WA Document Detail	1	HEADER	ZXY_PDIB_SWA_DOC_GRID1 ZXY_PDIB_LWA_DOC_GRID1	Header Header data
Web Application document list	WA List Header	1	HEADER	ZXY_PDIB_SWA_LIST_GRID1 ZXY_PDIB_LWA_LIST_GRID1	Header Header data
Workflow status popup	Workflow status	1	HEADER	ZXY_PDIB_SWA_STAT_GRID1 ZXY_PDIB_LWC_STAT_GRID1	Header Header data
Archive log	Header data for archive log	1	HEADER	ZXY_PDIB_SARC_LOG_GRID1 ZXY_PDIB_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses/substatuses for Invoice Block/Cancelation:

Statuses

Status	Status Description
ER	In error or incomplete
NN	New or ready for processing
OK	Processed

The following system document substatuses are available for invoice block/cancelation:

Substatus	Description
IC	Invoice cancelation
NB	Set new payment block

Substatus	Description
UB	Remove payment block

Text types

PROCESS DIRECTOR ships with this system text type for Invoice Block/Cancelation:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

Master Data Maintenance (MD)

The following standard master data types are currently available:

Master data type	SAP Object type	Package
Asset master	/EBY/PDMD_ASSET	/EBY/PDMD_ASSET
Cost center	/EBY/PDMD_COSTCTR	/EBY/PDMD_COSTCENTER
Customer master	/EBY/PDMD_CUSTMA	/EBY/PDMDCM
G/L account	/EBY/PDMD_GLACC	/EBY/PDMD_GLACCOUNT
Profit center	/EBY/PDMD_PRCTR	/EBY/PDMD_PROFITCENTER
Vendor master	/EBY/PDMD_VENDOMA	/EBY/PDMD_VENDORMASTER

This chapter provides reference information for these master data types.

Preset classes for Master Data Maintenance

The following preset classes are available for Master Data Maintenance in the PROCESS DIRECTOR standard. You can also use general preset classes that are available for all process types.

Asset Master

Preset business area from cost center

`/EBY/CL_PDMDAS_PRESET_BUSAREA`

Presets the business area from the entered cost center.

Preset vendor name

`/EBY/CL_PDMDAS_PRESET_VENDNAME`

Presets the vendor name.

Cost Centers

TEMPLATE: Preset header

`/EBY/CL_PDMDCC_PRESET_TEMPL`

This preset serves as a template for your own coding to preset header data in a cost center maintenance request.

Customer Master

Preset: IBAN number from bank account

`/EBY/CL_PDMDCM_PRESET_BNK2IBAN`

Presets the IBAN number from the entered bank account.

Preset: SWIFT number from bank account

`/EBY/CL_PDMDCM_PRESET_BNK2SWFT`

Presets the SWIFT number from the entered bank account.

Preset: Bank account from IBAN number

`/EBY/CL_PDMDCM_PRESET_IBAN2BANK`

Presets the bank account number from the entered IBAN number.

G/L Accounts

Presets based on company code

`/EBY/CL_PDMDGL_CCODE_PRESETS`

Presets data based on the entered company code.

Preset language from chart of account

`/EBY/CL_PDMDGL_PRESET_COA_LANG`

Presets the language from the entered Chart of Accounts.

Vendor Master

IBAN preset

`/EBY/CL_PDMDVM_EVT_PRESET_IBAN`

Presets the IBAN number based on the entered country and bank key. Add this preset class for the field `/EBY/IBAN`.

Bank details preset: name, SWIFT code

`/EBY/CL_PDMDVM_PRESET_BANKDETS`

Presets the bank name and SWIFT code based on the entered country and bank key. Add this preset class for the fields `BANKA` and `SWIFT`.

Preset: Bank account from IBAN number

`/EBY/CL_PDMDVM_PRESET_IBAN2BNK`

Presets the bank country, key, account number and control key based on the entered IBAN number. Add this preset class for the fields `BANKS`, `BANKL`, `BANKN` and `BKONT`.

Checks for Master Data Maintenance

The following check events are available for master data in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Asset Master

Simulate asset master creation/change/delete

`/EBY/CL_PDMDAS_EVT_CHK_POST`

Simulates the creation/change/deletion of the asset master data in SAP and returns all SAP messages generated during the simulation.

Cost Centers

Simulate cost center creation/change/delete

`/EBY/PDMDCC_EVT_CHK_POST`

Simulates the creation/change/deletion of the cost center in SAP and returns all SAP messages generated during the simulation.

Customer Master

Address check

`/EBY/CL_PDMDCM_EVT_CHK_ADDR`

Checks the address data.

Allowed alternative payer check

`/EBY/CL_PDMDCM_EVT_CHK_AAP`

Checks the allowed alternative payer.

Bank existence check

`/EBY/CL_PDMDCM_EVT_CHK_BANK`

Checks if the combination of country and bank key entered in the customer master maintenance request exists in the SAP system and returns a message if this is the case. This prevents existing bank data from being overwritten.

Check email address

`/EBY/CL_PDMDCM_EVT_CHK_EMAIL`

Checks if there is an email address in the customer master, and if yes, checks whether the email address is valid or not.

Check IBAN validity

`/EBY/CL_PDMDCM_EVT_CHK_IBAN`

Uses the SAP function `CHECK_IBAN` to check whether the IBAN number entered in the customer master maintenance request is valid and returns a message if this is not the case.

Check withholding tax

`/EBY/CL_PDMDCM_EVT_CHK_WHT`

Checks the withholding tax.

Sales area existence check

`/EBY/CL_PDMDCM_EVT_CHK_SAREA`

Checks if the sales area entered in the customer master maintenance request exists in the SAP system.

G/L Accounts

Check deletions flags - block is required

`/EBY/CL_PDMDGL_EVT_CHK_DELETE`

Checks whether deletion flags are activated for the document. This prevents the processor or approver from processing the document without looking at the **Block/Deletion** tab and missing deletion flags that have been set.

Check duplicates for same language

`/EBY/CL_PDMDGL_EVT_CHK_DPL_LNG`

Checks whether duplicate keywords and translations (G/L account descriptions) have been defined in PROCESS DIRECTOR for the same language.

Check G/L account posting

`/EBY/CL_PDMDGL_EVT_CHK_POST`

Simulates the creation/change/deletion of the G/L account in SAP and returns all SAP messages generated during the simulation.

Cost element existence check

`/EBY/CL_PDMDGL_EVT_CHK_COSTELM`

Checks whether the cost element entered in the G/L account maintenance request exists in the SAP system.

Profit Centers

Simulate profit center creation/change

`/EBY/PDXY_EVT_CHK_PRCRT`

Simulates the creation/change/deletion of the profit center in SAP and returns all SAP messages generated during the simulation.

Vendor Master

Address check

`/EBY/CL_PDMDVM_EVT_CHK_ADDR`

Checks the address data.

Bank existence check

`/EBY/CL_PDMDVM_EVT_CHK_BANK`

Checks if the combination of country and bank key entered in the vendor master maintenance request exists in the SAP system and returns a message if this is the case. This prevents existing bank data from being overwritten.

Check email address

`/EBY/CL_PDMDVM_EVT_CHK_EMAIL`

Checks if there is an email address in the vendor master, and if yes, checks whether the email address is valid or not.

Check IBAN validity

`/EBY/CL_PDMDVM_EVT_CHK_IBAN`

Uses the SAP function `CHECK_IBAN` to check whether the IBAN number entered in the vendor master maintenance request is valid and returns a message if this is not the case.

Check vendor master data for mandatory fields

`/EBY/CL_PDMDVM_EVT_CHK_POST`

Simulates vendor master data posting based on the standard SAP account group field status customizing.

Check permitted alternative payee

`/EBY/CL_PDMDVM_EVT_CHK_PAP`

Checks the permitted alternative payee.

Check withholding tax

`/EBY/CL_PDMDVM_EVT_CHK_WHT`

Checks the withholding tax.

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

Master data type	Logical level	Package	Interface structure	Customer Include
Cost center	Header data	/EBY/PDMD_COSTCENTER	/EBY/PDMDCC_SHDR_IF	CI_EBY_PDMDCC_SHDR_DATA
Cost center	Collection of cost centers	/EBY/PDMD_PROFITCENTER	/EBY/PDMDCC_SCCC_IF	CI_EBY_PDMDCC_SCCC_DATA
Profit center	Header data	/EBY/PDMD_PROFITCENTER	/EBY/PDMDPC_SHDR_IF	CI_EBY_PDMDPC_SHDR_DATA

Master data type	Logical level	Package	Interface structure	Customer Include
Vendor master	Header data	/EBY/PDMD_VENDORMASTER	/EBY/PDMDVM_SHDR_IF	CI_EBY_PDMDVM_SHDR_DATA CI_EBY_PDMDVM_SHDR_DISP
Vendor master	Bank details	/EBY/PDMD_VENDORMASTER	/EBY/PDMDVM_SBANK_DATA	CI_EBY_PDMDVM_SBANK_DATA
G/L account	Header data	/EBY/PDMD_GLACCOUNT	/EBY/PDMDGL_SHDR_DISP	CI_EBY_PDMDGL_SHDR_DISP
G/L account	Keywords	/EBY/PDMD_GLACCOUNT	/EBY/PDMDGL_SKYW_DATA	CI_EBY_PDMDGL_SKYW_DATA
G/L account	Translation	/EBY/PDMD_GLACCOUNT	/EBY/PDMDGL_STRL_DATA	CI_EBY_PDMDGL_STRL_DATA

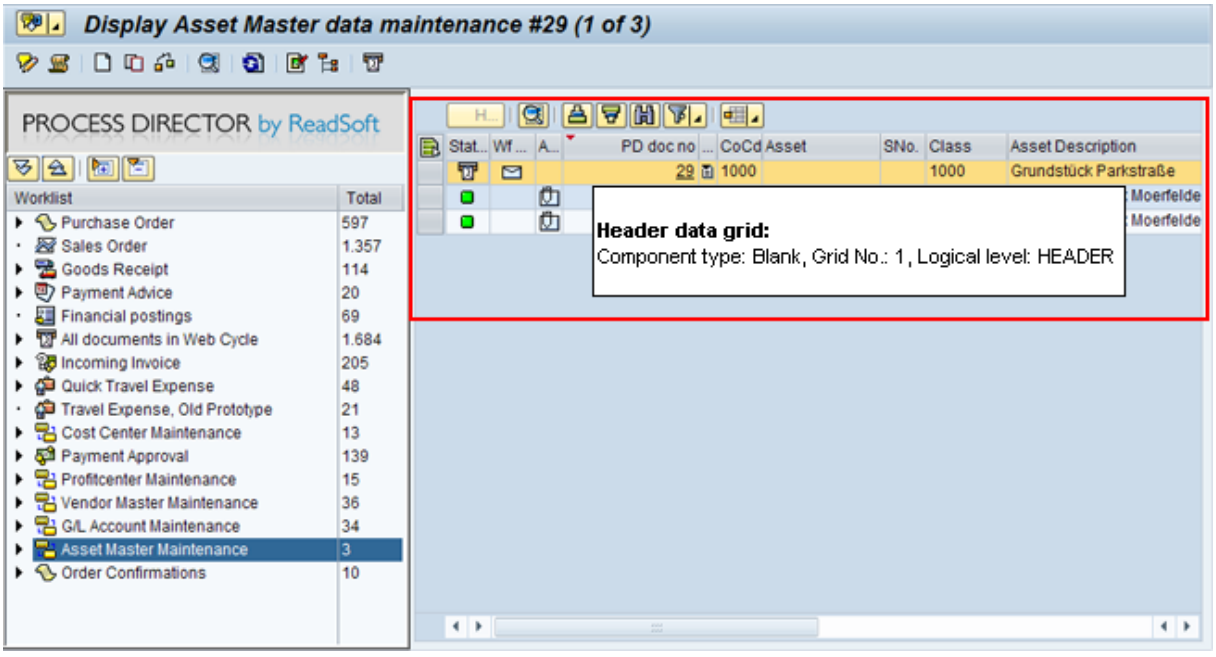
View model customization

Asset Master Maintenance

This section provides detailed information about which settings to enter when creating a customer view model for Asset Master Data Maintenance. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

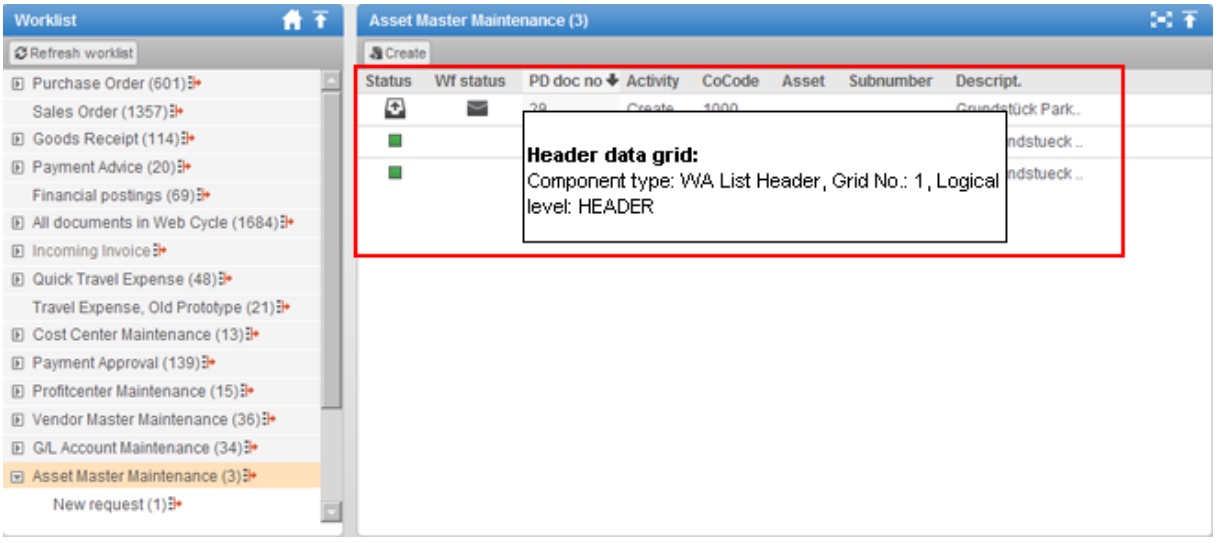
- SAP document overview list:



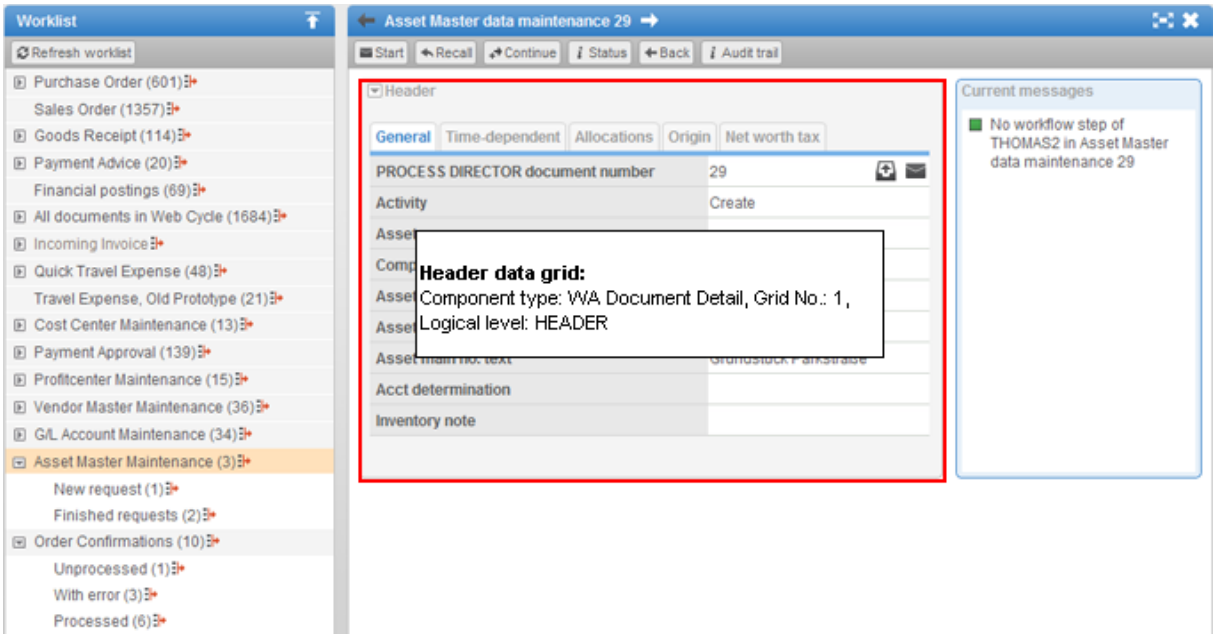
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

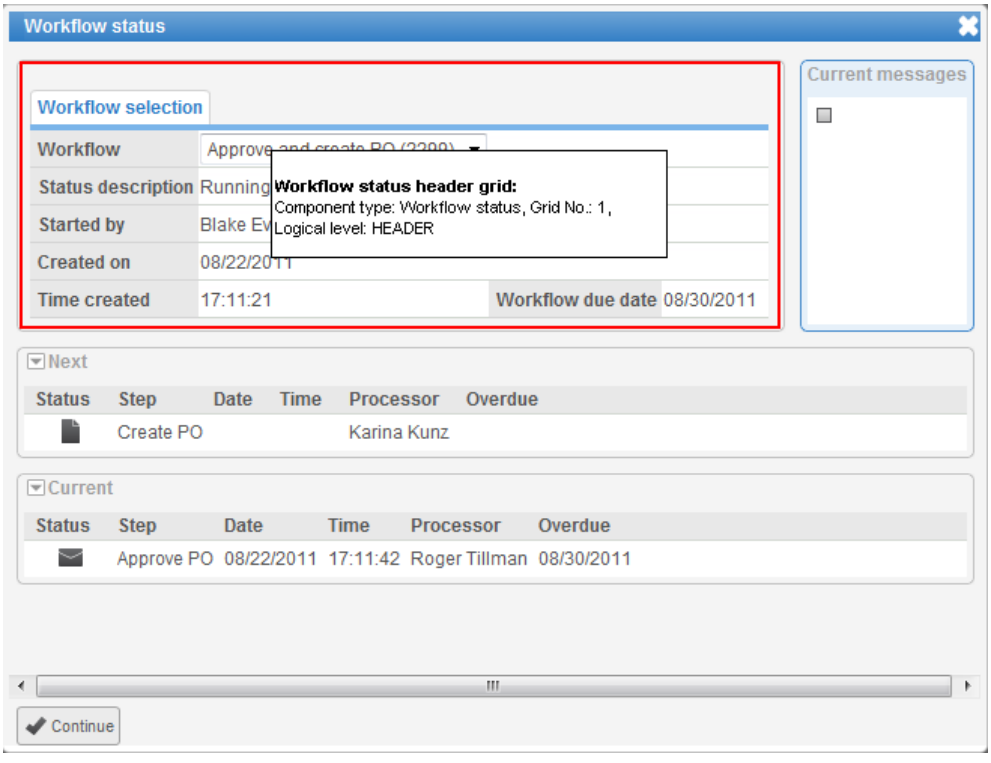
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

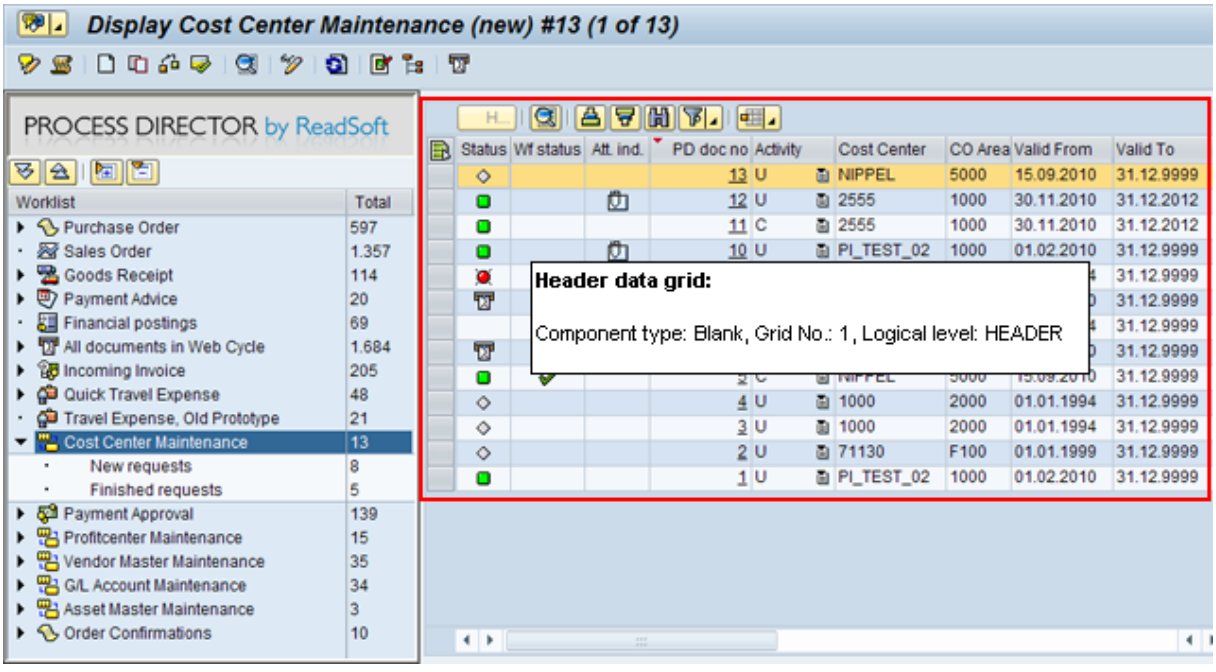
Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDMDAS_SGRID1 ZXY_PDMDAS_LGRID1	Header Header data
WA Document Detail	1	HEADER	ZXY_PDMDAS_SWA_DOC_GRID1 ZXY_PDMDAS_LWA_DOC_GRID1	Header Header data
WA List Header	1	HEADER	ZXY_PDMDAS_SWA_LIST_GRID1 ZXY_PDMDAS_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDMDAS_SWA_STAT_GRID1 ZXY_PDMDAS_LWC_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDMDAS_SARC_LOG_GRID1 ZXY_PDMDAS_LARC_LOG_GRID1	Header Header data

Cost Center Maintenance

This section provides detailed information about which settings to enter when creating a customer view model for Cost Center Maintenance. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

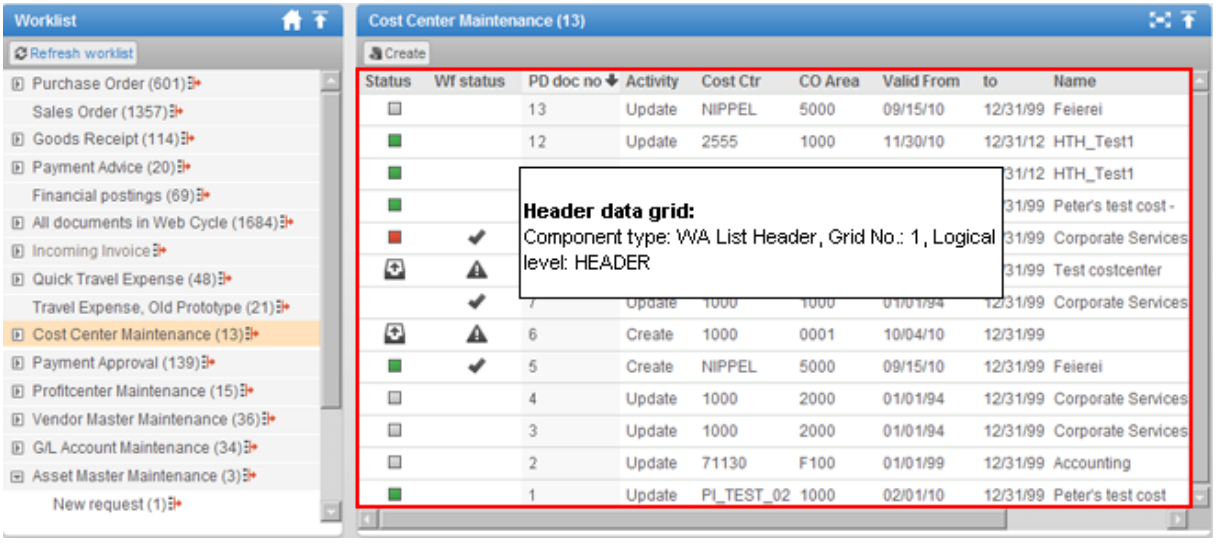
- SAP document overview list:



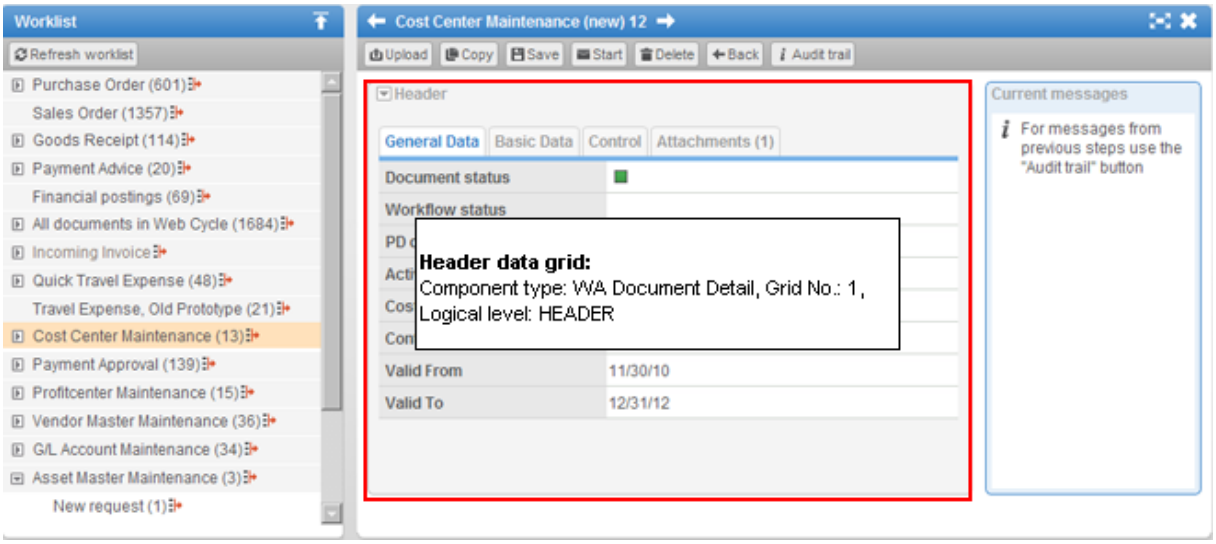
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

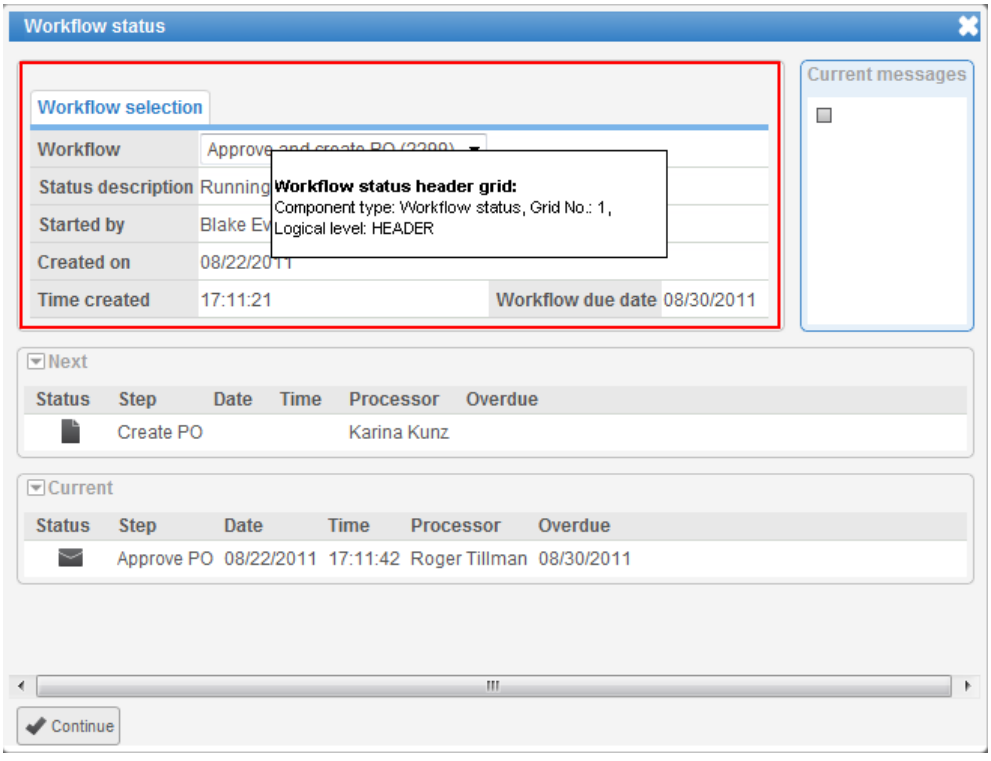
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

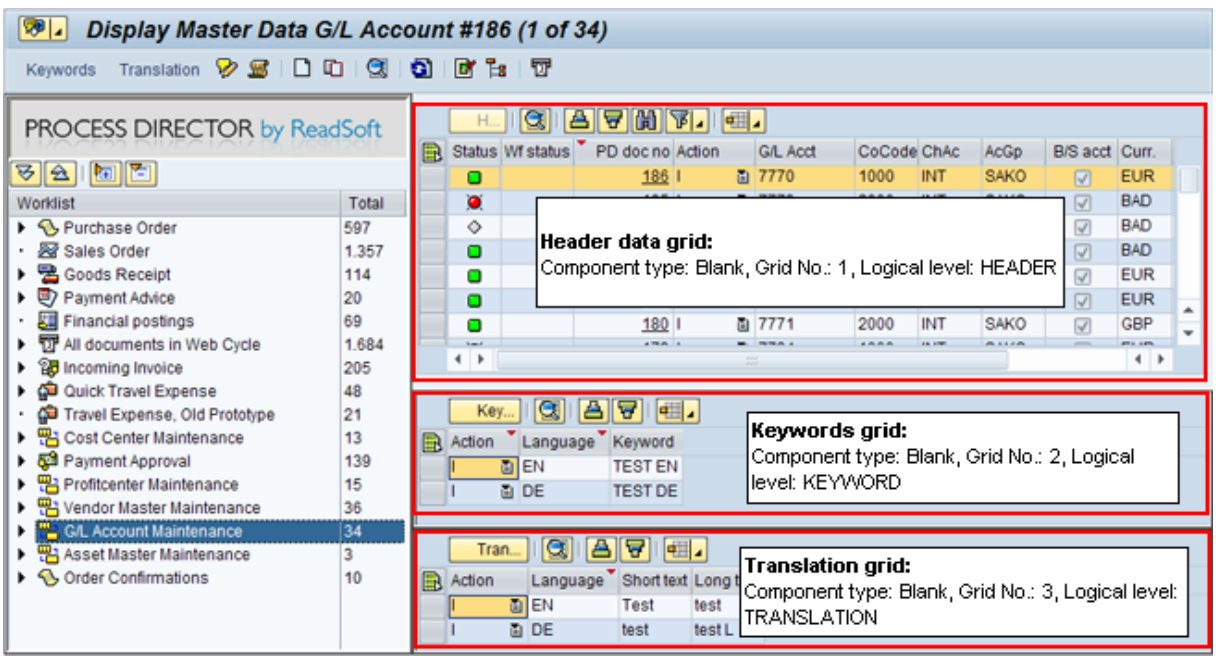
Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDMDCC_SGRID1 ZXY_PDMDCC_LGRID1	Header Header data
WA Document Detail	1	HEADER	ZXY_PDMDCC_SWA_DOC_GRID1 ZXY_PDMDCC_LWA_DOC_GRID1	Header Header data
WA List Header	1	HEADER	ZXY_PDMDCC_SWA_LIST_GRID1 ZXY_PDMDCC_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDMDCC_SWA_STAT_GRID1 ZXY_PDMDCC_LWC_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDMDCC_SARC_LOG_GRID1 ZXY_PDMDCC_LARC_LOG_GRID1	Header Header data

G/L Account Maintenance

This section provides detailed information about which settings to enter when creating a customer view model for G/L Account Maintenance. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

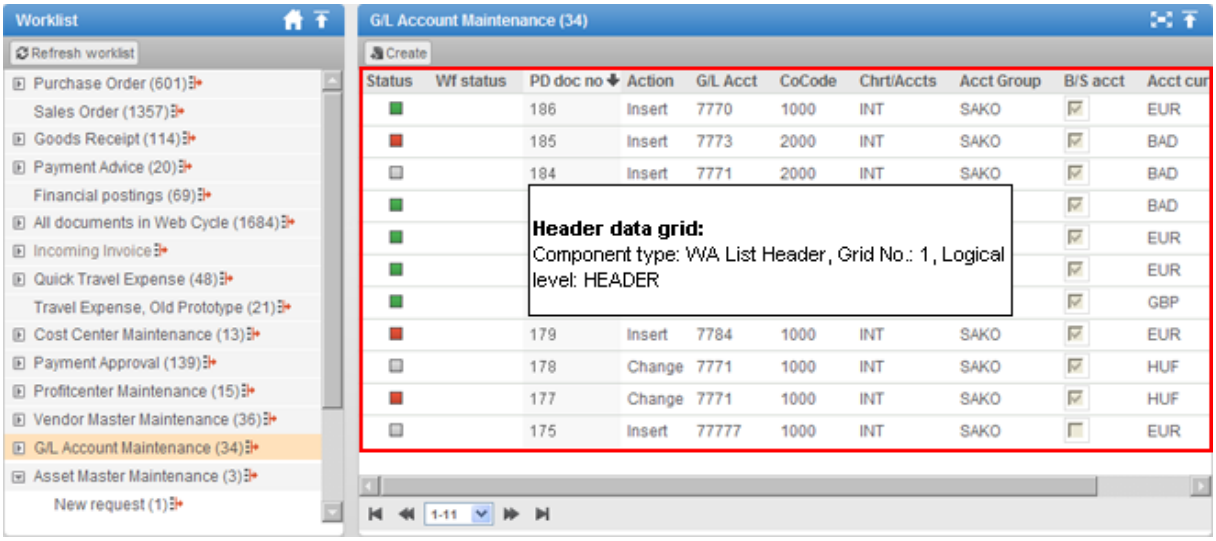
- SAP document overview list:



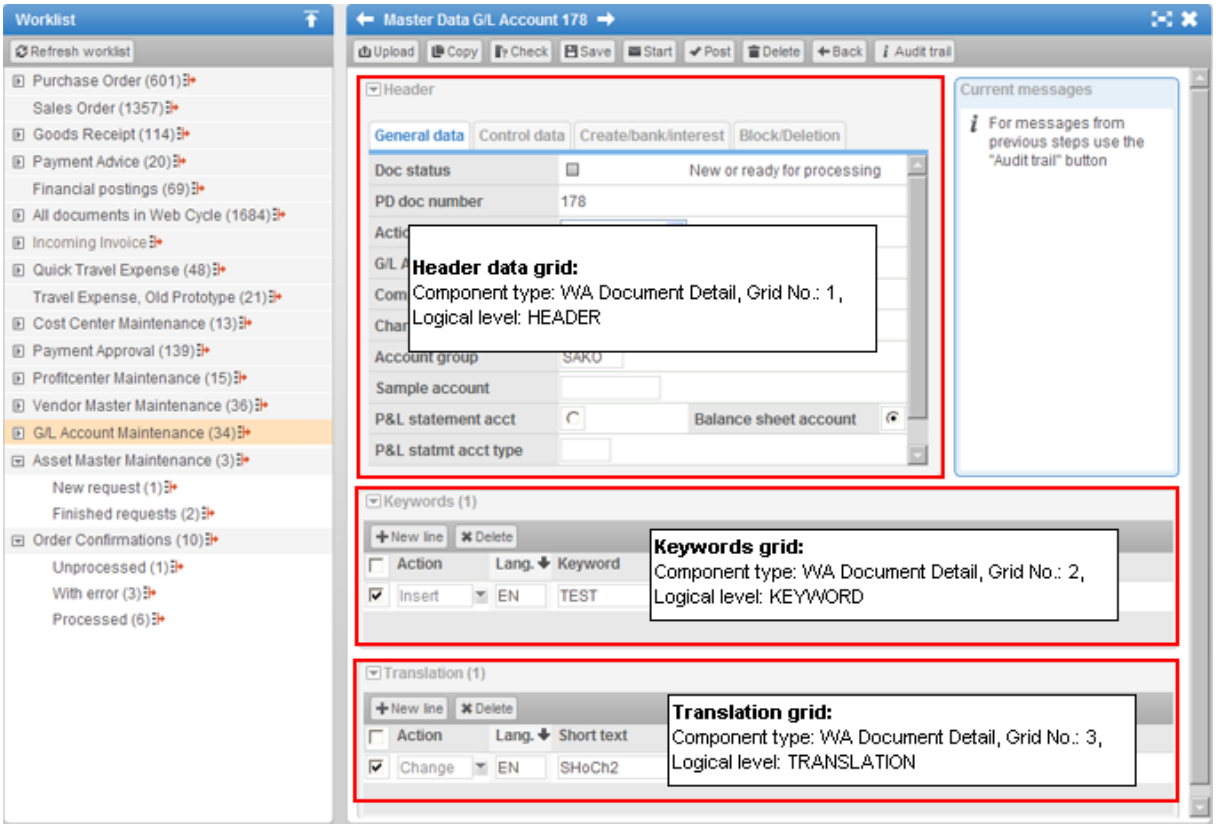
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

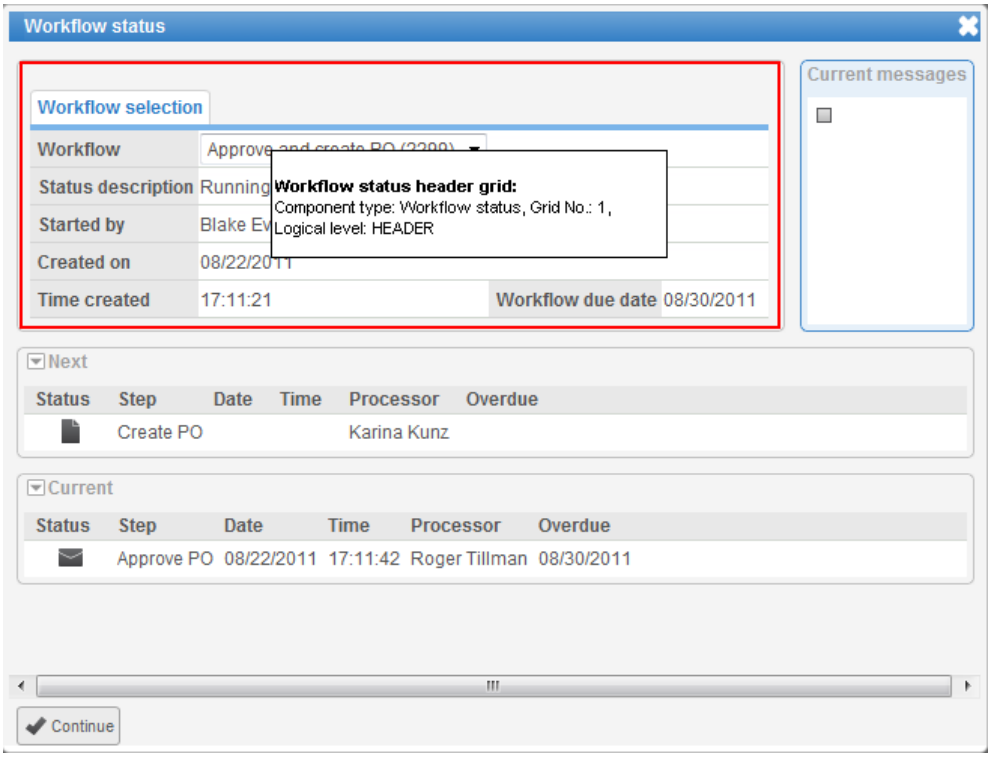
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDMDGL_SGRID1 ZXY_PDMDGL_LGRID1	Header Header data
	2	KEYWORD	ZXY_PDMDGL_SGRID2 ZXY_PDMDGL_LGRID2	Keywords Keywords
	3	TRANSLATION	ZXY_PDMDGL_SGRID3 ZXY_PDMDGL_LGRID3	Translation Translation
WA Document Detail	1	HEADER	ZXY_PDMDGL_SWA_DOC_GRID1 ZXY_PDMDGL_LWA_DOC_GRID1	Header Header data
WA Document Detail	2	KEYWORD	ZXY_PDMDGL_SWA_DOC_GRID2 ZXY_PDMDGL_LWA_DOC_GRID2	Keywords Keywords
WA Document Detail	3	TRANSLATION	ZXY_PDMDGL_SWA_DOC_GRID3 ZXY_PDMDGL_LWA_DOC_GRID3	Translation Translation

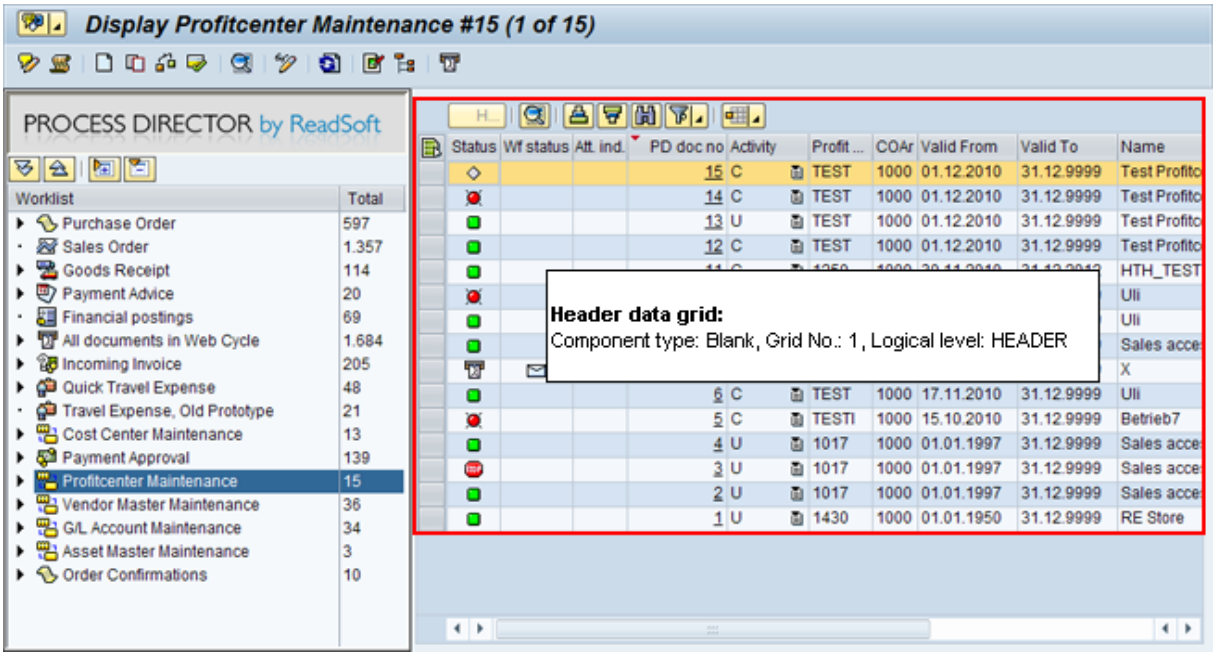
Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
WA List Header	1	HEADER	ZXY_PDMDGL_SWA_LIST_GRID1 ZXY_PDMDGL_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDMDGL_SWA_STAT_GRID1 ZXY_PDMDGL_LWC_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDMDGL_SARC_LOG_GRID1 ZXY_PDMDGL_LARC_LOG_GRID1	Header Header data

Profit Center Maintenance

This section provides detailed information about which settings to enter when creating a customer view model for Profit Center Maintenance. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

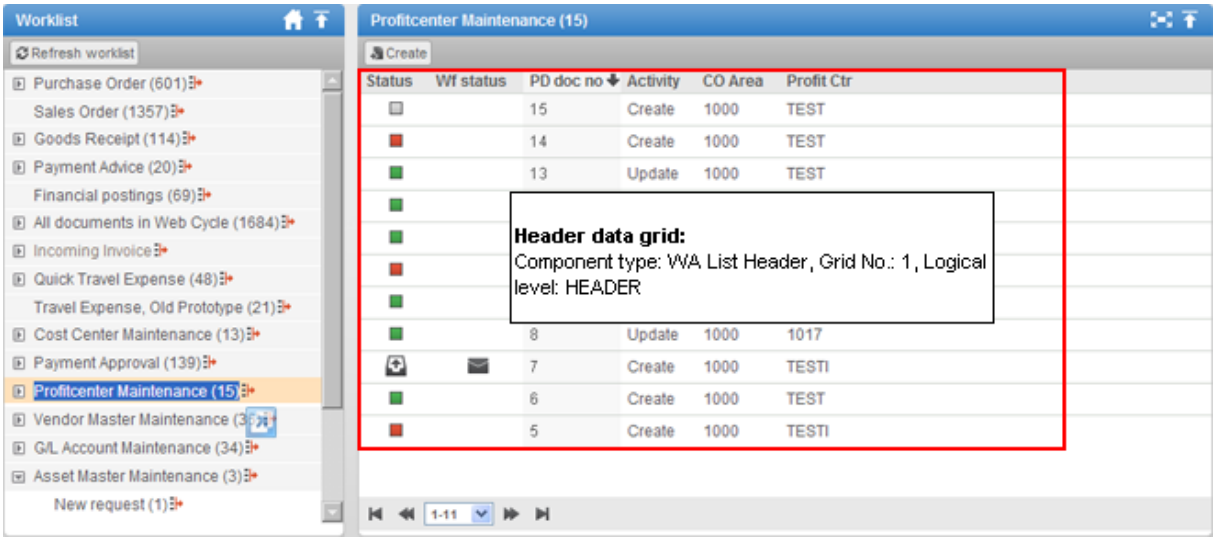
- SAP document overview list:



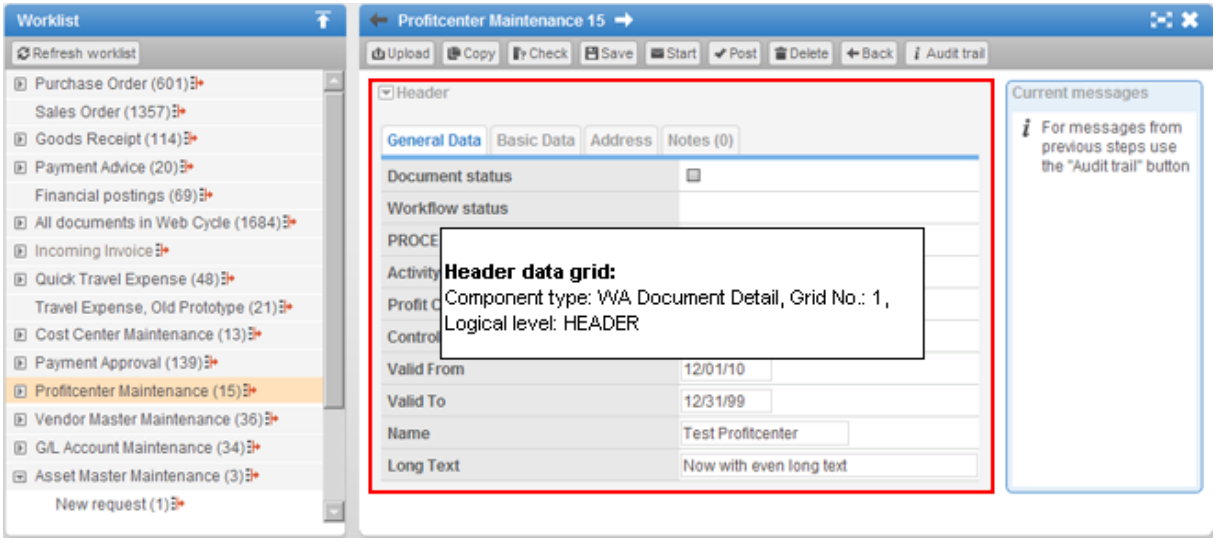
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

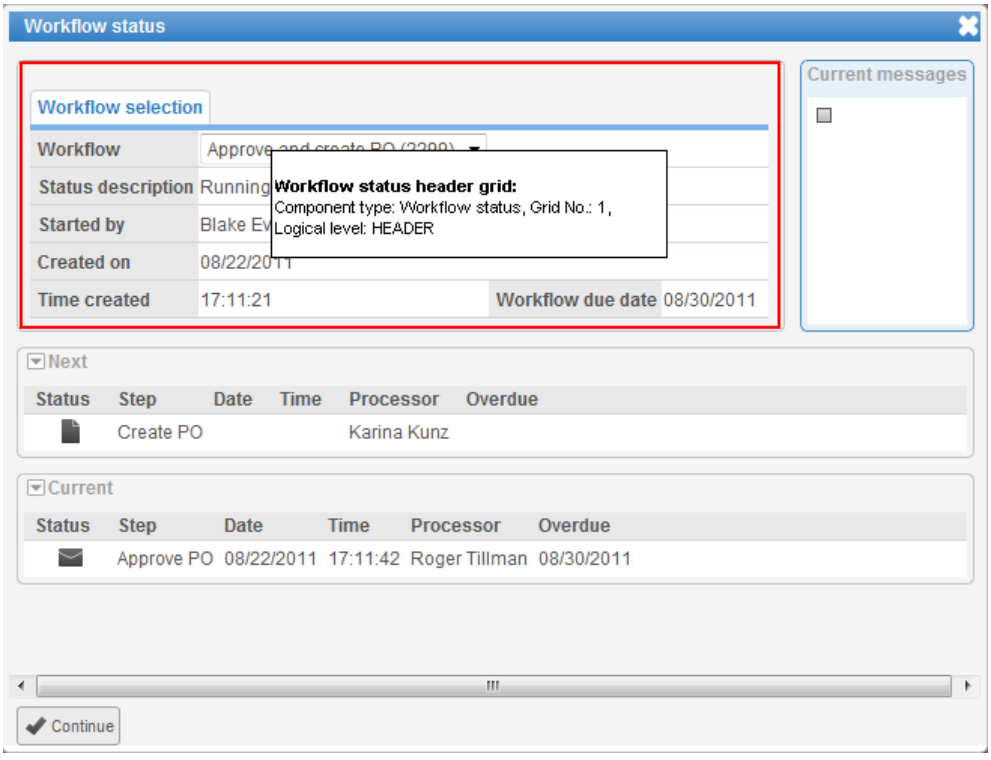
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

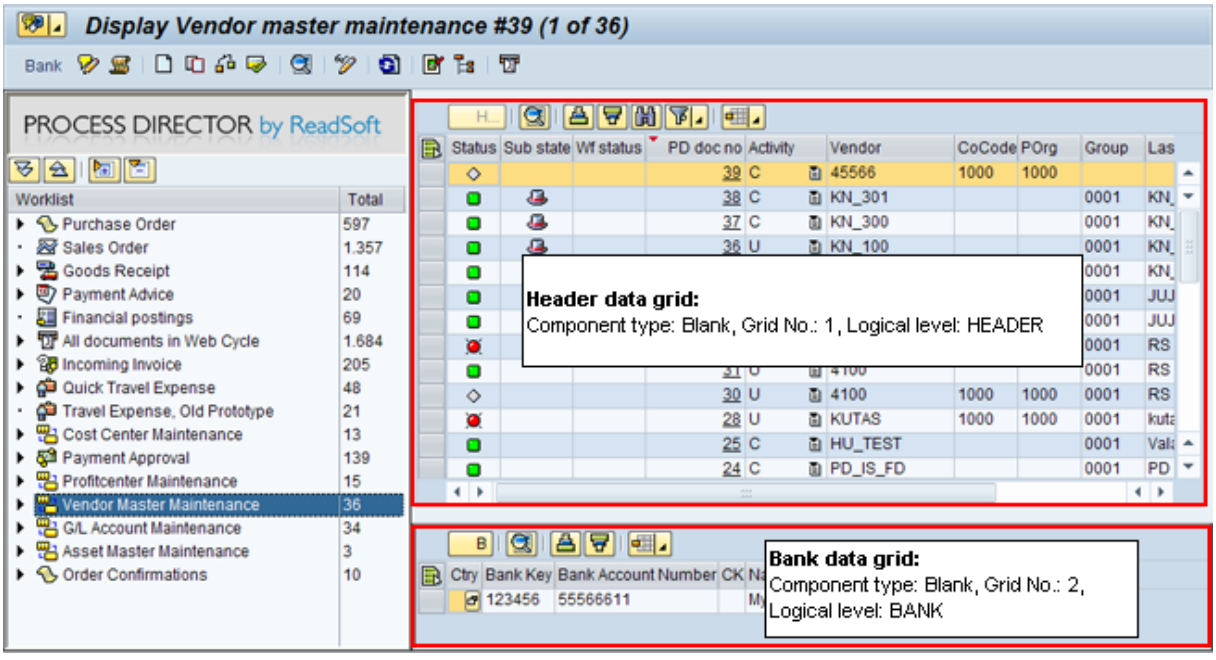
Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDMDPC_SGRID1 ZXY_PDMDPC_LGRID1	Header Header data
WA Document Detail	1	HEADER	ZXY_PDMDPC_SWA_DOC_GRID1 ZXY_PDMDPC_LWA_DOC_GRID1	Header Header data
WA List Header	1	HEADER	ZXY_PDMDPC_SWA_LIST_GRID1 ZXY_PDMDPC_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDMDPC_SWA_STAT_GRID1 ZXY_PDMDPC_LWC_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDMDPC_SARC_LOG_GRID1 ZXY_PDMDPC_LARC_LOG_GRID1	Header Header data

Vendor Master Maintenance

This section provides detailed information about which settings to enter when creating a customer view model for Vendor Master Data Maintenance. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

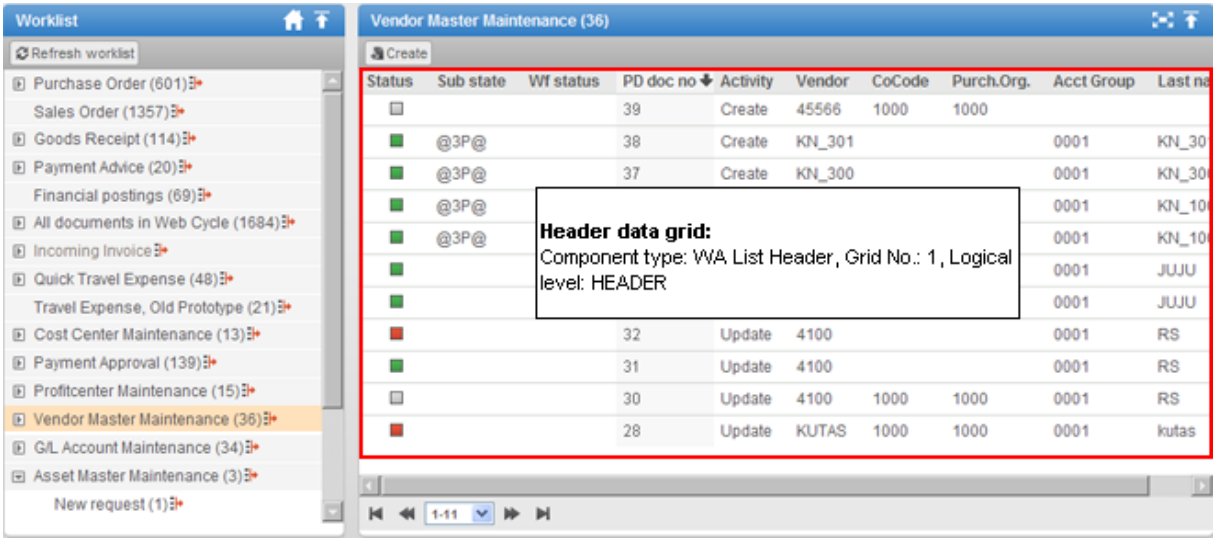
- SAP document overview list:



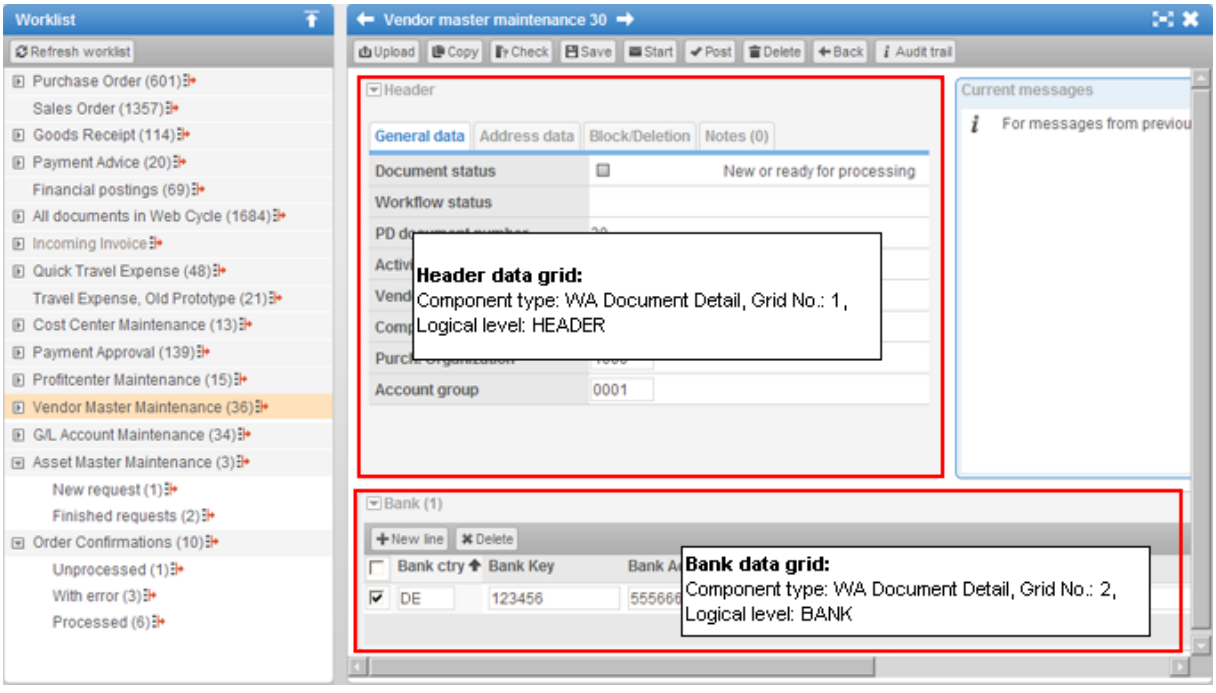
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

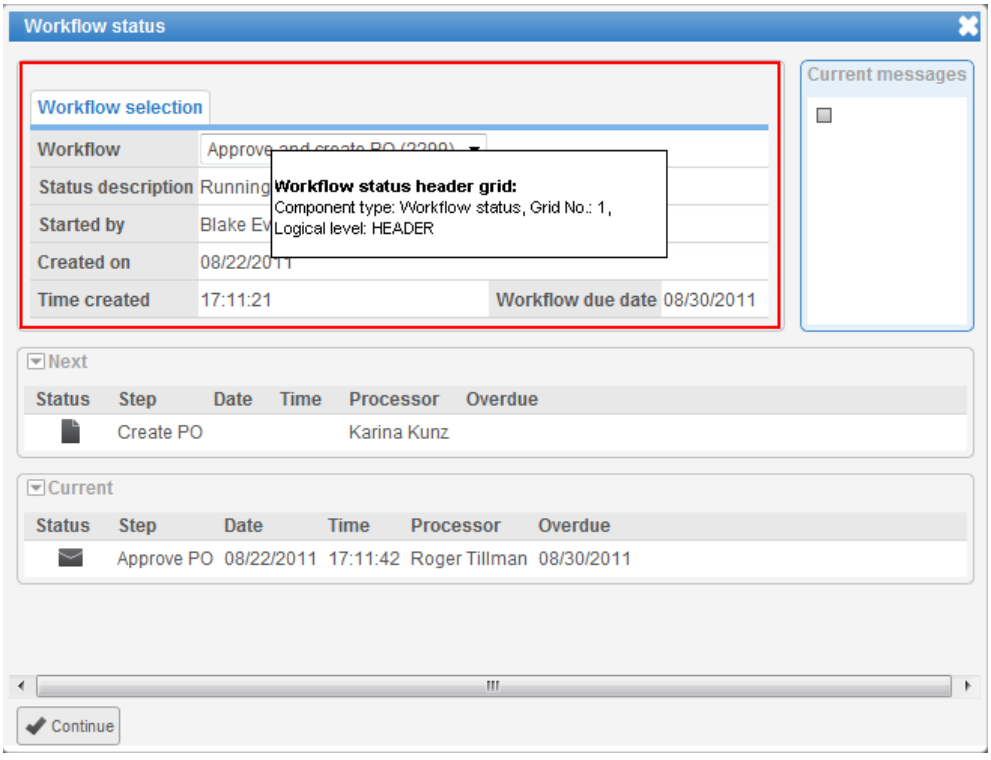
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDMDVM_SGRID1 ZXY_PDMDVM_LGRID1	Header Header data
	2	BANK	ZXY_PDMDVM_SGRID2 ZXY_PDMDVM_LGRID2	Bank Bank
WA Document Detail	1	HEADER	ZXY_PDMDVM_SWA_DOC_GRID1 ZXY_PDMDVM_LWA_DOC_GRID1	Header Header data
WA Document Detail	2	BANK	ZXY_PDMDVM_SWA_DOC_GRID2 ZXY_PDMDVM_LWA_DOC_GRID2	Bank Bank
WA List Header	1	HEADER	ZXY_PDMDVM_SWA_LIST_GRID1 ZXY_PDMDVM_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDMDVM_SWA_STAT_GRID1 ZXY_PDMDVM_LWC_STAT_GRID1	Header Header data

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
Header data for archive log	1	HEADER	ZXY_PDMDVM_SARC_LOG_GRID1 ZXY_PDMDVM_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses for all Master Data Maintenance types:

Status	Status Description
ER	In error or incompleted
NN	New or ready for processing
OK	Processed
WR	Postable with warning(s) (Customer and Vendor master data only)
XX	Rejected

The following system document substatuses are available for specific master data maintenance types:

Vendor Master

Substatus	Description
LA	General data posted
LB	Accounting data posted
LF	Posting completed
LM	Purchasing data posted

Customer Master

Substatus	Description
LA	General data posted
LB	Accounting data posted
LF	Posting completed
LS	Sales area data posted

Asset Master

Substatus	Description
CE	Create - External number range
CI	Create - Internal number range
D	Delete
U	Update

Text types

PROCESS DIRECTOR ships with this system text type for Master Data Maintenance:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

User exits / BAdIs

BAdI Posting (G/L Account Maintenance)

This user exit / BAdI currently provides the interface methods `BEFORE_POST` and `AFTER_POST` that you can implement to add your own customer logic before or after a G/L Account Maintenance document is posted.

BEFORE_POST method

Calling sequence

This method is called from the action `POST_DOC Post document`. It runs immediately *before* document posting occurs.

Method signature

Type	Parameter	Description
	`VALUE(IR_OBJECT)`	Generic header
	`IS_BO_HDR`	Data fields of a PD base object
	`IS_MDGL_HDR`	G/L account header data
	`IT_MDGL_KYW`	G/L account keywords data
	`IT_MDGL_TRL`	G/L account translations data
	`CS_ACCOUNT_COA`	G/L account master record: Chart of Accounts
	`CT_ACCOUNT_NAMES`	G/L account master record: Descriptions (Table)
	`CT_ACCOUNT_KEYWORDS`	G/L account master record: Key Word (Table)
	`CT_ACCOUNT_CCODES`	G/L account master record: Company Code (Table)

AFTER_POST method

Calling sequence

This method is called from the action `POST_SUCCESS Posting succeeded`. It runs after the document status `Posted` has been set.

Note: If document posting fails, `AFTER_POST` is *not* called.

Method signature

Type	Parameter	Description
	`(VALUE)IR_OBJECT`	Generic header
	`IS_ACCOUNT_COA`	G/L account master record: Chart of Accounts
	`IT_ACCOUNT_NAMES`	G/L account master record: Descriptions (Table)
	`IT_ACCOUNT_KEYWORDS`	G/L account master record: Key Word (Table)
	`IT_ACCOUNT_CCODES`	G/L account master record: Company Code (Table)
	`CS_BO_HDR`	Data fields of a PD base object
	`CS_MDGL_HDR`	PROCESS DIRECTOR generated structure
	`CT_MDGL_KYW`	PROCESS DIRECTOR generated table type
	`CT_MDGL_TRL`	PROCESS DIRECTOR generated table type

BAdI Posting (Vendor Master Data Maintenance)

This user exit / BAdI currently provides the interface methods `BEFORE_POST` and `AFTER_POST` that you can implement to add your own customer logic before or after a Vendor Master Data Maintenance document is posted.

BEFORE_POST method

Calling sequence

This method is called from the action `POST_DOC Post document`. It runs immediately *before* document posting occurs.

Method signature

Type	Parameter	Description
	`VALUE(IR_OBJECT)`	Generic header
	`IS_BO_HDR`	Data fields of a PD base object
	`IS_MDVM_HDR`	PROCESS DIRECTOR generated structure
	`IT_MDVM_BANK`	PROCESS DIRECTOR generated table type
	`CT_IDOIC_DATA`	Table type for EDIDD (IDoc Data Records)

AFTER_POST method

Calling sequence

This method is called from the action `POST_SUCCESS Posting succeeded`. It runs after the document status `Posted` has been set.

Note: If document posting fails, `AFTER_POST` is *not* called.

Method signature

Type	Parameter	Description
	`(VALUE)IR_OBJECT`	Generic header
	`IT_IDOC_DATA`	
	`CT_MDVM_BANK`	PROCESS DIRECTOR generated table type
	`CT_MDVM_HDR`	PROCESS DIRECTOR generated structure
	`CS_BO_HDR`	Data fields of a PD base object

Order Confirmations (OR)

SAP Object type: `/EBY/PDOR`

Package: `/EBY/PDOR`

This chapter provides reference information for Order Confirmations.

Checks for Order Confirmations

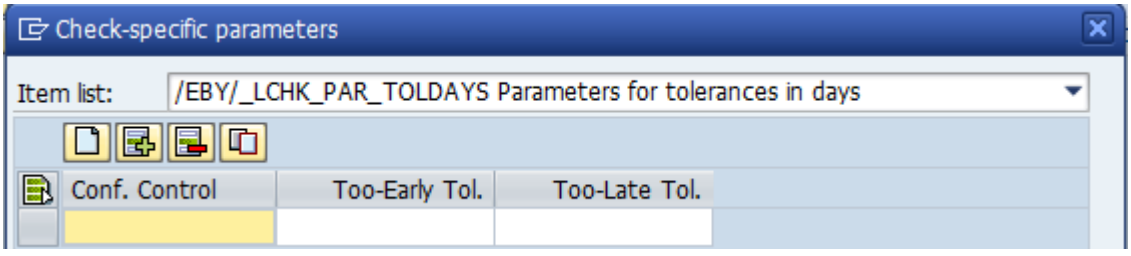
The following check events are available for order confirmations in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check confirmed delivery date

`/EBY/CL_PDOR_EVT_CHK_DATE`

Checks the delivery date given in the order confirmation against the PO.

In the initialization parameters, you can specify tolerances for specific confirmation control keys.



Check confirmed Incoterms

`/EBY/CL_PDOR_EVT_CHK_INCO`

Checks the Incoterms given in the order confirmation against the PO.

Check confirmed price

`/EBY/CL_PDOR_EVT_CHK_PRICE`

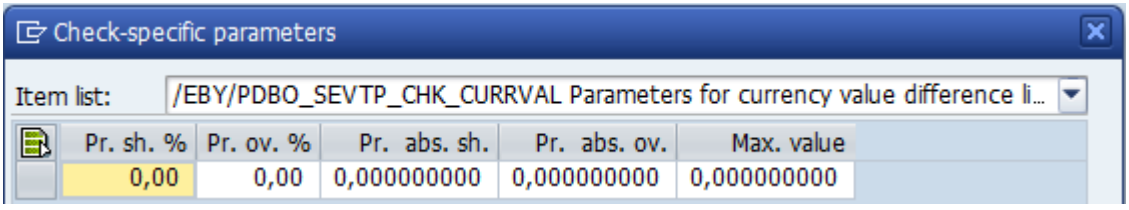
Checks the price given in the order confirmation against the PO.

In the initialization parameters, you can define the following rules:

- value shortfall in %
- value overrun in %

- absolute value shortfall
- absolute value overrun
- limit for standard line item value

If one of the defined rules is broken, a message is generated.



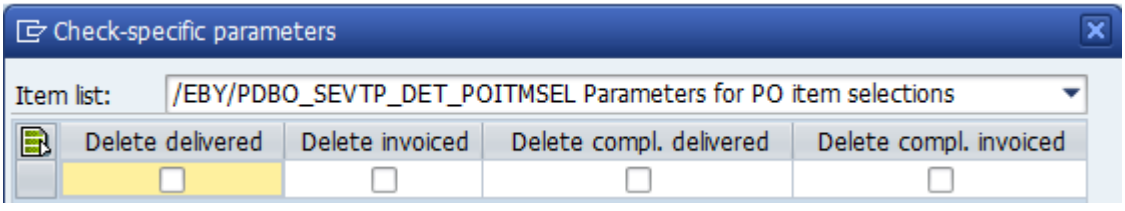
Check confirmed quantity

`/EBY/CL_PDOR_EVT_CHK_QTY`

Checks the quantity given in the order confirmation against the PO.

Overdelivery and underdelivery tolerances are taken from the PO.

In the initialization parameters, you can exclude delivered and invoiced items.



Initialization parameter	Description
Delete delivered	Excludes items that have been partially delivered.
Delete invoiced	Excludes items that have been partially invoiced.
Delete compl. delivered	Excludes items that have been completely delivered.
Delete compl. invoiced	Excludes items that have been completely invoiced.

Check whether the data can be processed

`/EBY/CL_PDOR_EVT_CHK_PROC`

Simulates the creation of the order confirmation in SAP and returns all SAP messages generated during the simulation.

Determinations for Order Confirmations

These determinations are available for Order Confirmations in the PROCESS DIRECTOR standard:

Purchase order line item assignment

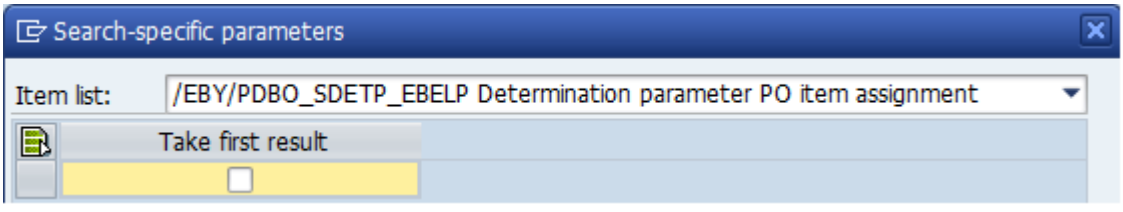
`/EBY/CL_PDOR_DET_EBELP_INIT`

Determines the purchase order line item number by comparing (depending on the defined searches):

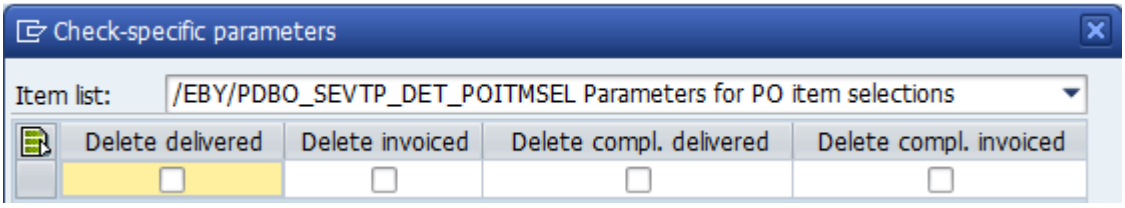
Search type	Search name
The vendor material number, line item number and quantity given on the order confirmation with the undelivered order quantity on the purchase order.	`/EBY/CL_PDOR_DET_EBELP_IDNLF`
The material number, line item number and quantity given on the order confirmation with the undelivered order quantity on the purchase order.	`/EBY/CL_PDOR_DET_EBELP_MAT`
The price and line item number given on the order confirmation with the price on the purchase order.	`/EBY/CL_PDOR_DET_EBELP_PRICE`
The quantity and line item number given on the order confirmation with the quantity on the purchase order.	`/EBY/CL_PDOR_DET_EBELP_QTY`

An exact match takes preference over a partial delivery.

In the search parameters, you can specify that the system should use the first match found.



You can also exclude delivered and invoiced items.



Initialization parameter	Description
Delete delivered	Excludes items that have been partially delivered.
Delete invoiced	Excludes items that have been partially invoiced.
Delete compl. delivered	Excludes items that have been completely delivered.
Delete compl. invoiced	Excludes items that have been completely invoiced.

PO item proposal as determination

`/EBY/CL_PDOR_DET_ITEMPROP_INIT`

Adds line items from the purchase order to the order confirmation. Only line items that have confirmation control set in the **Confirmations** tab in the PO can be added. Uses the search **Item proposal from PO** (`/EBY/CL_PDOR_DET_EBELP_ITEMPROP`).

Material determination

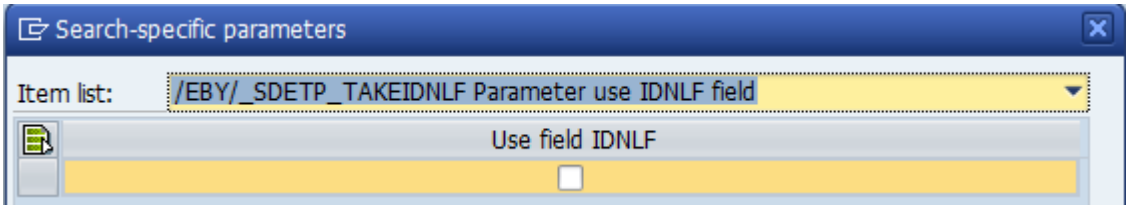
`/EBY/CL_PDOR_DET_MAT_INIT`

Determines the material number by checking whether the material number given on the order confirmation is (depending on the defined searches):

Search type	Search name
The material number from the material master data	`/EBY/CL_PDOR_DET_MAT_MARA`
A manufacturer part number maintained in the material master data	`/EBY/CL_PDOR_DET_MAT_MANU`

Search type	Search name
An EAN maintained in the material master data	`/EBY/CL_PDOR_DET_MAT_EAN`
A vendor material number maintained in the table `EINA`	`/EBY/CL_PDOR_DET_MAT_EINA`
The vendor material number given on the purchase order	`/EBY/CL_PDOR_DET_MAT_VENDOR`

In the initialization parameters, you can specify that the field `IDNLF` should be used:



Purchase Order determination

`/EBY/CL_PDOR_DET_PO_INIT`

Verifies the purchase order number given on the order confirmation by searching the `EKKO` (purchasing document header) table for this number. Uses the search **Search PO in table EKKO** (`/EBY/CL_PDOR_DET_PO_EKKO`).

Vendor determination

`/EBY/CL_PDOR_DET_VENDOR_INIT`

Verifies the vendor based on (depending on the defined searches):

Search type	Search name
The account number given on the order confirmation	`/EBY/CL_PDOR_DET_VENDOR_ACC`
The account number given on the purchase order	`/EBY/CL_PDOR_DET_VENDOR_ACCPO`
The vendor number given on the order confirmation	`/EBY/CL_PDOR_DET_VENDOR_LFA1`
The vendor number given on the purchase order	`/EBY/CL_PDOR_DET_VENDOR_PO`
The VAT number given on the order confirmation	`/EBY/CL_PDOR_DET_VENDOR_STCEG`

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

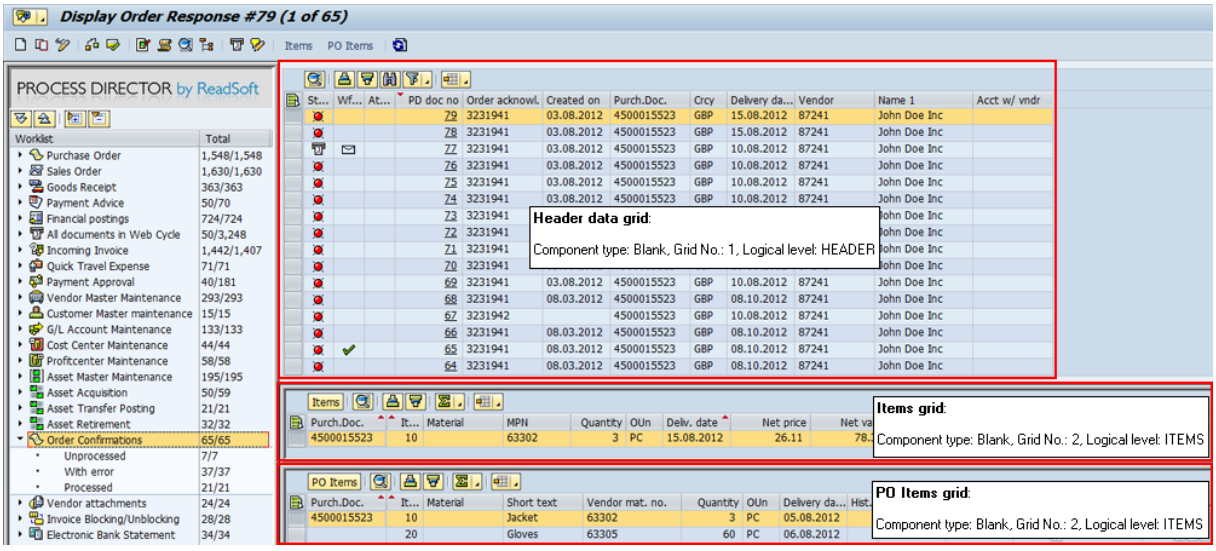
Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/PDOR_MODEL	/EBY/PDOR_SHDR_IF	CI_EBY_PDOR_SHDR_DATA CI_EBY_PDOR_SHDR_DISP
Item data	/EBY/PDOR_MODEL	/EBY/PDOR_SITM_IF	CI_EBY_PDOR_SITM_DATA CI_EBY_PDOR_SITM_DISP
PO item data	/EBY/PDOR_MODEL	/EBY/PDOR_SEKPO_IF	CI_EBY_PDOR_SEKPO_DATA CI_EBY_PDOR_SEKPO_DISP

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Order Confirmations. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

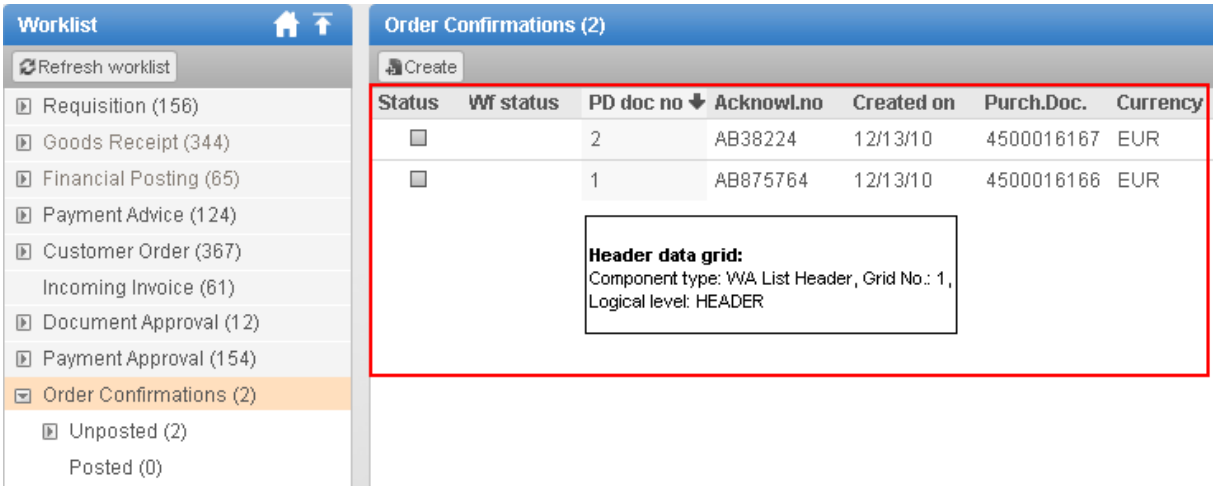
- SAP document overview list:



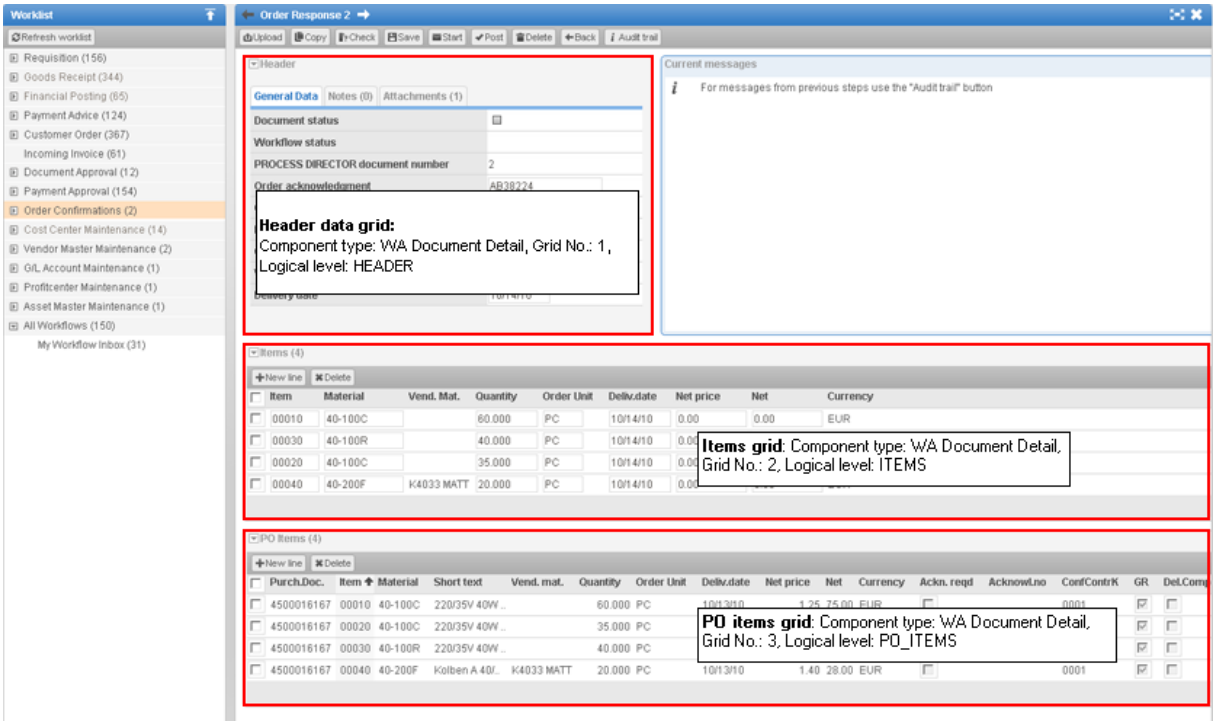
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

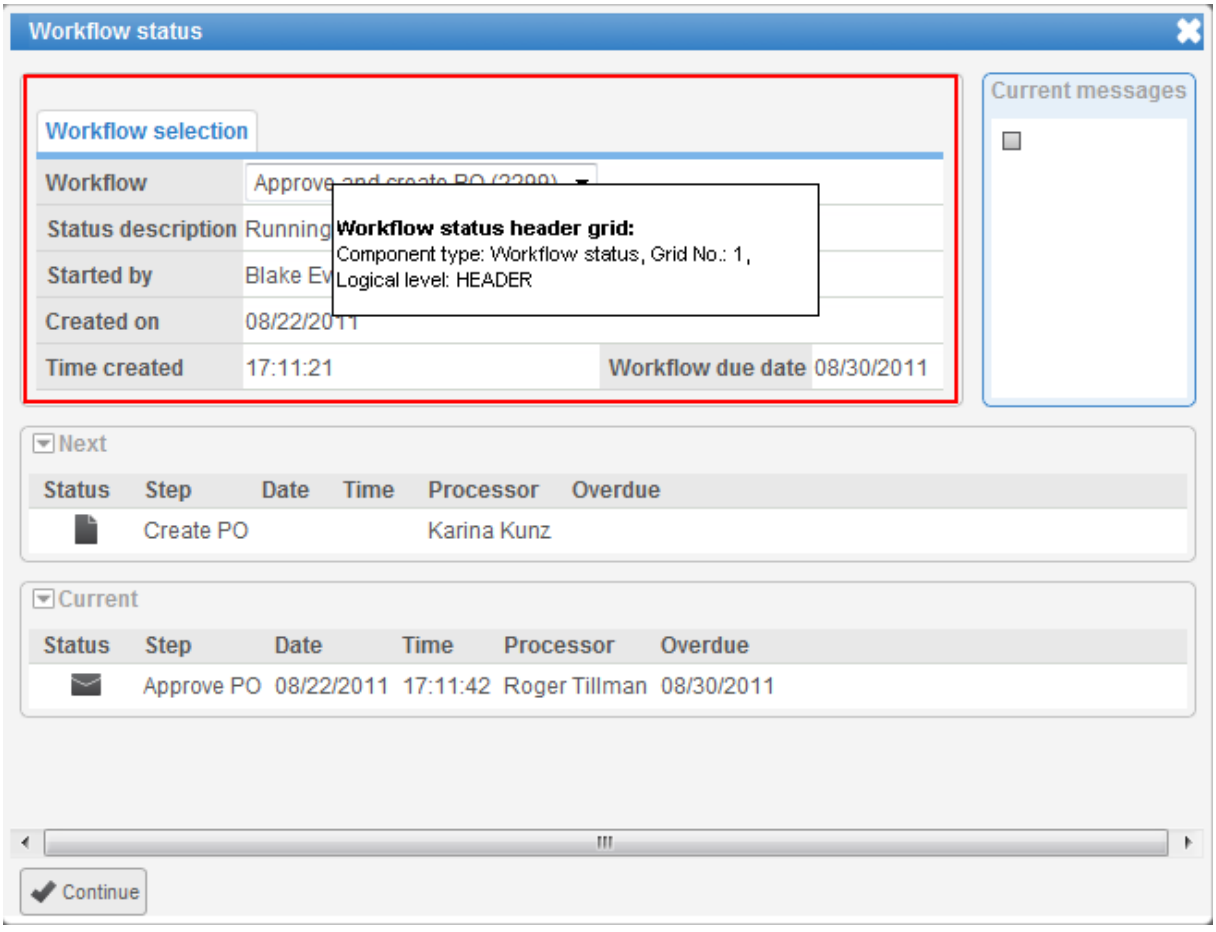
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDOR_SGRID1 ZXY_PDOR_LGRID1	Header Header data
	2	ITEMS	ZXY_PDOR_SGRID2 ZXY_PDOR_LGRID2	Items Items
	3	PO_ITEMS	ZXY_PDOR_SGRID3 ZXY_PDOR_LGRID3	PO Items PO Items
WA Document Detail	1	HEADER	ZXY_PDOR_SWA_DOC_GRID1 ZXY_PDOR_LWA_DOC_GRID1	Header Header data

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
WA Document Detail	2	ITEMS	ZXY_PDOR_SWA_DOC_GRID2 ZXY_PDOR_LWA_DOC_GRID2	Items Items
WA Document Detail	3	PO_ITEMS	ZXY_PDOR_SWA_DOC_GRID3 ZXY_PDOR_LWA_DOC_GRID3	PO Items PO Items
WA List Header	1	HEADER	ZXY_PDOR_SWA_LIST_GRID1 ZXY_PDOR_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDOR_SWA_STAT_GRID1 ZXY_PDOR_LWA_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDOR_SARC_LOG_GRID1 ZXY_PDOR_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses for Order Confirmations:

Status	Status Description
AP	Approved
ER	In error or incomplete
NN	New or ready for processing
OK	Processed
XX	Rejected

Text types

PROCESS DIRECTOR ships with this system text type for Order Confirmations:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

User exits / BAdIs

BAdI Posting (Order Confirmations)

This user exit / BAdI currently provides the interface methods `BEFORE_POST` and `AFTER_POST` that you can implement to add your own customer logic before or after an Order Confirmation document is posted.

BEFORE_POST method

Calling sequence

This method is called from the action `POST_DOC Post document`. It runs immediately *before* document posting occurs.

Method signature

Type	Parameter	Description
	`IR_OBJECT`	Order Response (OR) object
	`IS_BO_HDR`	Data fields of a PD base object
	`IS_OR_HDR`	Order Response (OR) header
	`IT_ITEMS`	Order Response (OR) items
	`IT_PO_ITEMS`	Generic PO items data
	`CT_IDOIC_DATA`	IDoc data

AFTER_POST method

Calling sequence

This method is called from the action `POST_SUCCESS Posting succeeded`. It runs after the document status `Posted` has been set.

Note: If document posting fails, `AFTER_POST` is *not* called.

Method signature

Type	Parameter	Description
	`(VALUE)IR_OBJECT`	Order Response (OR) object
	`IT_IDOC_DATA`	IDoc data
	`IT_PO_ITEMS`	Generic PO items data
	`CS_BO_HDR`	Data fields of a PD base object
	`CS_OR_HDR`	Order Response (OR) header
	`CT_ITEMS`	Order Response (OR) items

Payment Advices (PA)

SAP Object type: `/EBY/PDPA`

Package: `/EBY/PDPA`

This chapter provides reference information for Payment Advices.

Preset classes for Payment Advices

The following preset classes are available for Payment Advices in the PROCESS DIRECTOR standard. You can also use general preset classes that are available for all process types.

Preset customer name

`/EBY/CL_PDPA_PRESET_CUSTOMER`

Presets the customer name from the customer number. Set this preset for the `NAME` field.

Preset item data from BSID

`/EBY/CL_PDPA_PRESET_ITEM_BSID`

Presets the following item data from the table `BSID`:

- Accounting document line item number
- Fiscal year
- Alternative account
- Debit/credit indicator

The customer number in the header data is also populated, if it is initial.

The company code and accounting document number must be known.

Using this preset can be faster than using open item determinations.

Preset payment amount from gross amount

`/EBY/CL_PDPA_PRESET_PAYMNT`

At header level, presets the payment amount from the gross amount, or the gross amount from the payment amount.

Updates items in the Electronic Bank Statement

`/EBY/CL_PDES_PRESET_ITEM`

Adds the Payment Advice number and customer number to line items in the Electronic Bank Statement.

Note: Although this preset is a `PDES` class, it must be added to the Payment Advices presets configuration, not the Electronic Bank Statement preset configuration.

Checks for Payment Advices

The following check events are available for Payment Advices in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check company code

`/EBY/CL_PDPA_EVT_CHK_COMP_CODE`

Checks whether the company code entered in the payment advice is maintained in SAP and is valid.

Check if item matches a customer open item

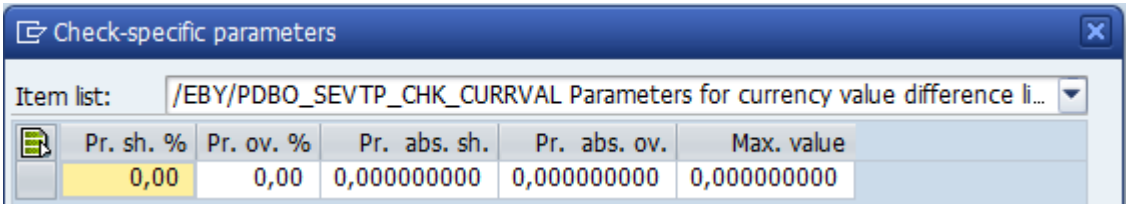
`/EBY/CL_PDPA_EVT_CHK_OPEN_ITEM`

Checks whether this line item has been billed to a customer and whether the line item is still open (not paid).

In the initialization parameters, you can define the following rules:

- value shortfall in %
- value overrun in %
- absolute value shortfall
- absolute value overrun
- limit for standard line item value

If one of the defined rules is broken, a message is generated.



Check integrity of header amounts

`/EBY/CL_PDPA_EVT_CHK_HDR_AMNT`

Checks whether the values in the total amount fields of the header are equal to the sum of the corresponding line item fields.

Check integrity of item amounts

`/EBY/CL_PDPA_EVT_CHK_ITM_AMNT`

Checks whether the values in the amount fields of the line items are correct (Payment amount should be equal to Gross amount - Discount amount - Deduction amount).

Check the customer number of the payment advice

`/EBY/CL_PDPA_EVT_CHK_CUSTOMER`

Checks whether the customer number entered in the payment advice is maintained in SAP for the given company code and is valid.

Determinations for Payment Advices

These determinations are available for Payment Advices in the PROCESS DIRECTOR standard.

Payment Advice customer determination

`/EBY/CL_PDPA_DET_CUST_INIT`

Determines the customer number based on (depending on the defined searches):

Search type	Search name
KNA1 (Customer Master General Data)	`/EBY/CL_PDPA_DET_CUST_KNA1`
KNB1 (Customer Master Company Code Data)	`/EBY/CL_PDPA_DET_CUST_KNB1`
KNVV (Customer Master Sales Data)	`/EBY/CL_PDPA_DET_CUST_KNVV`
Amount. This search first tries to match the payment amount and gross amount. If no match is found, it then considers discounts, local currency, and statistical values to find a match.	`/EBY/CL_PDPA_DET_CUST_AMOUNT`
Accounting document number	`/EBY/CL_PDPA_DET_CUST_BELNR`
Document date	`/EBY/CL_PDPA_DET_CUST_BLDAT`
Posting date	`/EBY/CL_PDPA_DET_CUST_BUDAT`
Billing document number	`/EBY/CL_PDPA_DET_CUST_VBELN`
Reference document number	`/EBY/CL_PDPA_DET_CUST_XBLNR`
Reference document number from reference in item	`/EBY/CL_PDPA_DET_CUST_XR_XBLNR`

It searches in open items, cleared documents and parked documents to try and find the best match. Various search parameters are available.

Payment Advice item mapper determination

`/EBY/CL_PDPA_DET_ITEMPROP_INIT`

Determines the document number, item number and fiscal year in the line items by searching for (depending on the defined searches):

Search type	Search name
Amount. This search first tries to match the payment amount and gross amount. If no match is found, it then considers discounts, local currency, and statistical values to find a match.	`/EBY/CL_PDPA_DET_ITEMMAP_AMT`
Document number	`/EBY/CL_PDPA_DET_ITEMMAP_BELNR`
Document date	`/EBY/CL_PDPA_DET_ITEMMAP_BLDAT`
Posting date	`/EBY/CL_PDPA_DET_ITEMMAP_BUDAT`
Item text	`/EBY/CL_PDPA_DET_ITEMMAP_SGTXT`
Billing document	`/EBY/CL_PDPA_DET_ITEMMAP_VBELN`
Reference number	`/EBY/CL_PDPA_DET_ITEMMAP_XBLNR`

It searches in open items, cleared documents and parked documents to try and find the best match. Various search parameters are available.

Payment Advice item proposal determination

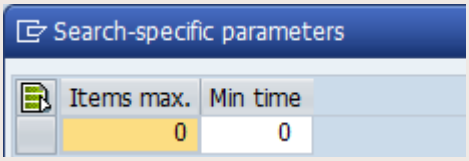
`/EBY/CL_PDPA_DET_ITEMPROP_INIT`

Retrieves open items for the company code and customer number entered in the document header and creates a corresponding line item for each open item. In the search parameters, you can specify the maximum number of items that should be retrieved.

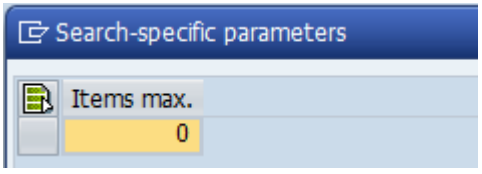
Depending on the defined searches, the following items are added:

Search type	Search name

Propose all items	`/EBY/CL_PDPA_DET_ITEMPROP`
	or
	`/EBY/CL_PDPA_DET_ITEMPROP_Z00`

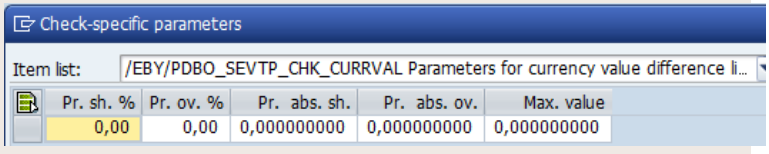
Currently due items. This means items for which the invoice due date is later than the payment date, and the posting date lies before the payment date by a specified minimum number of days, which is defined in the search parameters (**Min time**).  	`/EBY/CL_PDPA_DET_ITEMPROP_Z21`
Overdue items	`/EBY/CL_PDPA_DET_ITEMPROP_Z22`
Unique combination of items. This means items are only added if there is a single, unique combination of open items that matches the payment amount. Example for unique combination: Assume the following customer open items are available: ▪ Item 1 with the amount 100 ▪ Item 2 with the amount 200 ▪ Item 3 with the amount 150 ▪ Item 4 with the amount 250 If the payment amount is 250, two different combinations of items match the payment amount: ▪ Item 1 + Item 3 = 250 ▪ Item 4 = 250 If the payment amount is 600, there is a unique combination, because only one combination of items matches the payment amount: Item 2 + Item 3 + Item 4 = 600 (no other combination amounts to 600)	`/EBY/CL_PDPA_DET_ITEMPROP_Z231`

In the search parameters, you can specify the maximum number of items that should be retrieved:



Search parameters

Which of these parameters are available depends on the selected search.

Parameter type	Description
/EBY/PDPA_LEVTP_BS_FLD_FILTER	**Condition for BSAD and BSEG filtering** Enables you to specify conditions for the search. Only items that meet the specified criteria (or do not meet them, depending on the selected option) are included in the search. For example, particularly for cleared items, you may want to filter out items that are older than a specific date to exclude them from the search.
/EBY/PDBO_SEVTP_CHK_CURRVAL	**Parameters for currency value difference limits** Enables you to specify tolerances for currency values: ▪ value shortfall in % ▪ value overrun in % ▪ absolute value shortfall ▪ absolute value overrun ▪ limit for standard line item value If one of the defined rules is broken, a message is generated. 

Parameter type	Description
/EBY/PDPA_LEVTP_ITEM_TO_BS_FLD	**Mapping of BSID, BSAD and BSEG fields to PDPA items** Enables you to specify which fields in the SAP tables are compared with which PROCESS DIRECTOR Payment Advice fields, and in which order the fields are compared.
/EBY/PDBO_SEVTP_FIELD_PROP	**Field properties** Enables you to specify a minimum field length. If the field value is shorter than this length, it is not taken into account by the determination. If you check the **Wildcard** option, you can also specify a partial minimum length.

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/PDPA_MODEL	/EBY/PDPA_SHDR_IF	CI_EBY_PDPA_SHDR_DATA CI_EBY_PDPA_SHDR_DISP

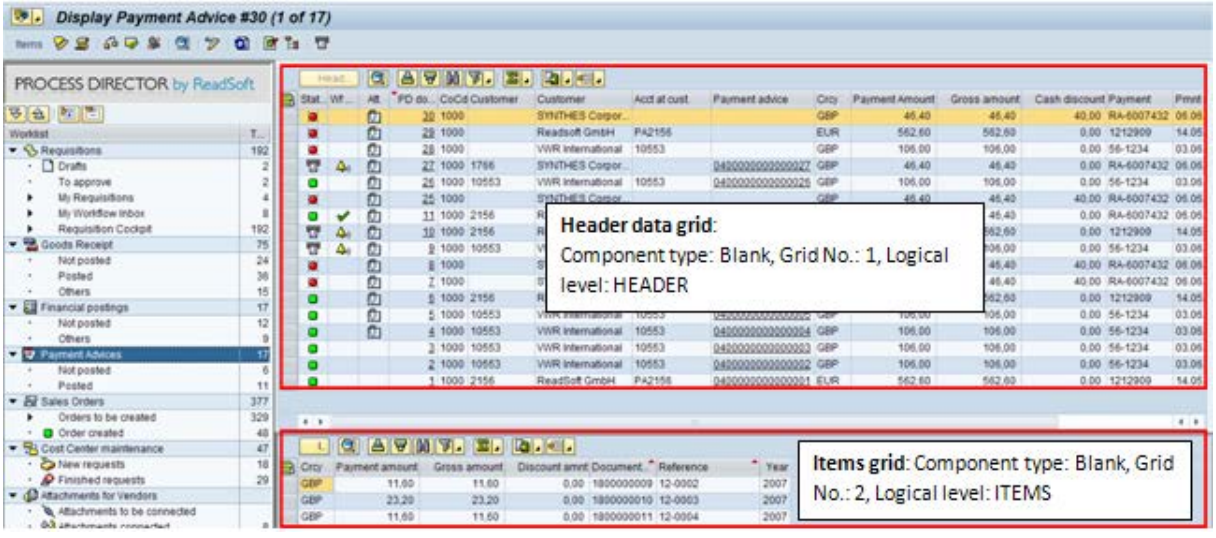
Logical level	Package	Interface structure	Customer Includes
Item data	/EBY/PDPA_MODEL	/EBY/PDPA_SITM_IF	CI_EBY_PDPA_SITM_DATA CI_EBY_PDPA_SITM_DISP

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Payment Advices. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

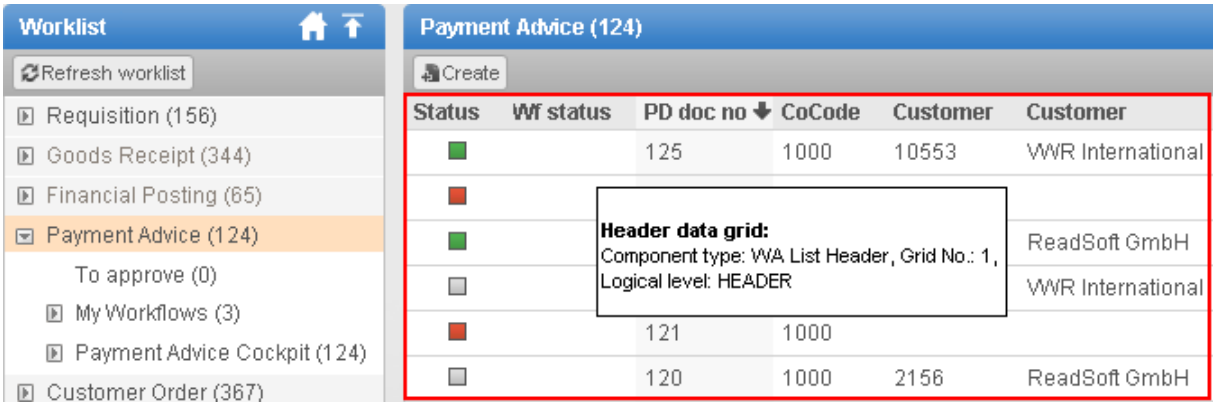
- SAP document overview list:



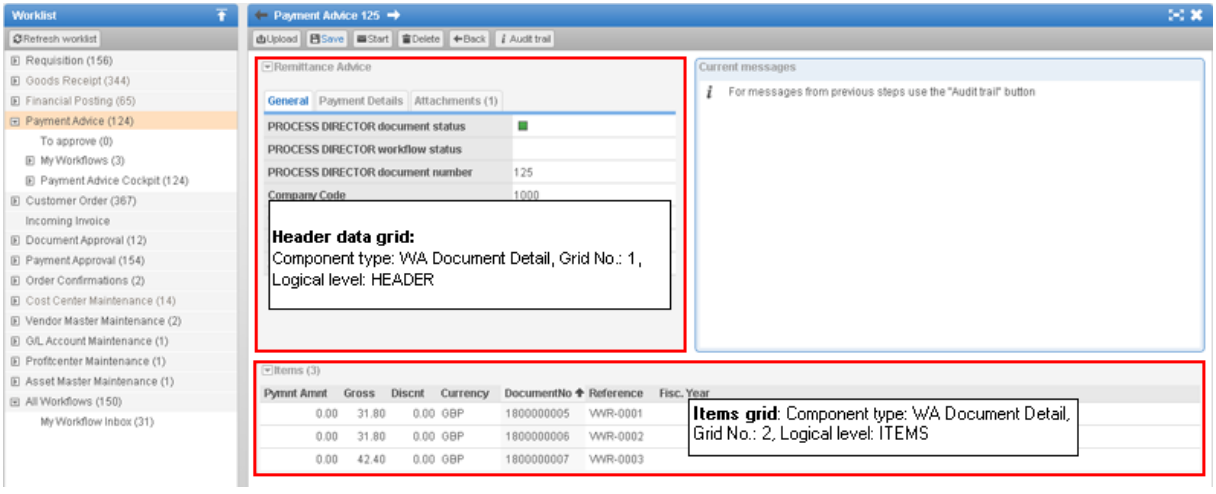
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction /EBY/PDVI_VDSTC.

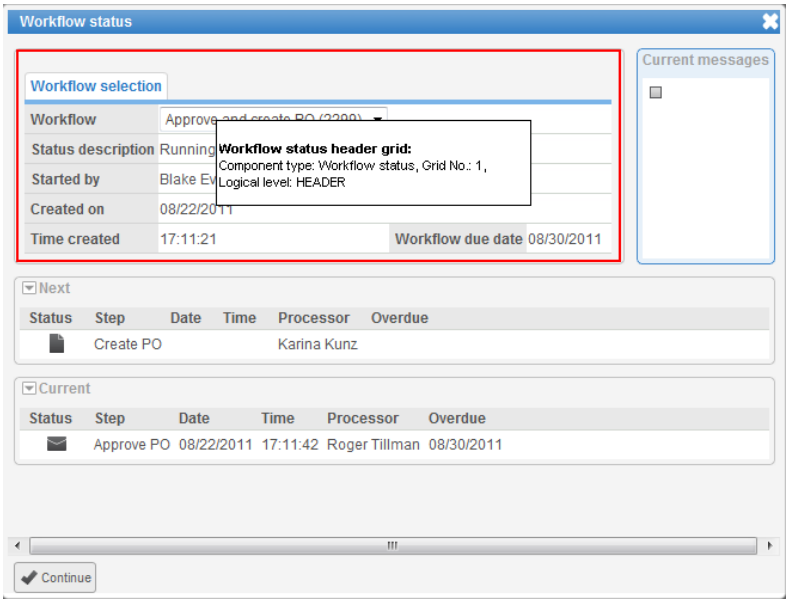
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDPA_SGRID1 ZXY_PDPA_LGRID1	Header Header data
	2	ITEMS	ZXY_PDPA_SGRID2 ZXY_PDPA_LGRID2	Items Items
WA Document Detail	1	HEADER	ZXY_PDPA_SWA_DOC_GRID1 ZXY_PDPA_LWA_DOC_GRID1	Remittance Advice Remittance Advice
WA Document Detail	2	ITEMS	ZXY_PDPA_SWA_DOC_GRID2 ZXY_PDPA_LWA_DOC_GRID2	Items Items
WA List Header	1	HEADER	ZXY_PDPA_SWA_LIST_GRID1 ZXY_PDPA_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDPA_SWA_STAT_GRID1 ZXY_PDPA_LWA_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDPA_SARC_LOG_GRID1 ZXY_PDPA_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses for Payment Advices:

Status	Status Description
ER	New or in error
OK	Correct
PO	Posted
XX	Rejected

User exits / BAdIs

BAdI Posting (Payment Advices)

This user exit / BAdI currently provides the interface methods `BEFORE_POST` and `AFTER_POST` that you can implement to add your own customer logic before or after a Payment Advice document is posted.

BEFORE_POST method

Calling sequence

This method is called from the action `POST_DOC Post document`. It runs immediately *before* document posting occurs.

Method signature

Type	Parameter	Description
	`VALUE(IR_OBJECT)`	Payment Advices object
	`IS_BO_HDR`	Data fields of a PD base object
	`IS_PA_HDR`	Payment Advices (PA) header data
	`IT_ITEMS`	Payment Advices (PA) items

Type	Parameter	Description
	IB_BACKGROUND	Background/online processing
	CT_BDCDATA	Batch Input Branch to Objects
	CC_TCODE	Transaction code
	CS_OPTIONS	Parameter string for runtime of CALL TRANSACTION USING ...

AFTER_POST method

Calling sequence

This method is called from the action `POST_SUCCESS Posting succeeded`. It runs after the document status `Posted` has been set.

Note: If document posting fails, `AFTER_POST` is *not* called.

Method signature

Type	Parameter	Description
	(VALUE)IR_OBJECT	Payment Advices object
	IT_BDCDATA	Batch Input Branch to Objects
	IC_TCODE	Transaction code
	IT_MESSAGES	Collecting messages in the SAP system
	IB_BACKGROUND	Background/online processing
	IS_OPTIONS	Parameter string for runtime of CALL TRANSACTION USING ...

Type	Parameter	Description
	II_RC	Return code after CALL TRANSACTION
	CS_BO_HDR	Data fields of a PD base object
	CS_PA_HDR	Payment Advices (PA) header data
	CT_ITEMS	Payment Advices (PA) items

Programs

The following program is available for use with Payment Advices:

- /EBY/PDPA_FINALIZE_DOC

  Checks if the open items of the payment advice document are already cleared and links the payment advice document to the clearing posting.

/EBY/PDPA_FINALIZE_DOC

Use the /EBY/PDPA_FINALIZE_DOC program to link SAP clearing documents to PROCESS DIRECTOR payment advice documents.

The program checks if the open items of the payment advice document are already cleared and links the payment advice document to the clearing posting. Items that are not assigned are ignored.

Payment Approvals (PR)

SAP Object type: /EBY/PDPR

Package: /EBY/PDPR

This chapter provides reference information for Payment Approvals.

Checks for Payment Approvals

The following checks are available for Payment Approvals in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check payment proposal was recreated

`/EBY/CL_PDPR_EVT_CHK_CREA_DATE`

Checks whether the payment proposal was deleted and recreated in SAP with the same identification. If this is the case, a warning message is issued.

Check payment block

`/EBY/CL_PDPR_EVT_CHK_PMT_BLOCK`

Checks whether payment blocks have been set in the payment proposal document.

Check payment proposal approved

`/EBY/CL_PDPR_EVT_CHK_PMT_PROPO`

Checks whether workflow approval has been completed for the entire payment proposal.

Check payment proposal approved for one PD document

`/EBY/CL_PDPR_EVT_CHK_PMT_PRP_1`

Checks whether workflow approval has been completed for a single PROCESS DIRECTOR document.

Process parameters for Payment Approvals

The following process parameters are available for Payment Approvals. You can also use the general process parameters that are available for all process types.

Process	Description
Archive the output of a print report (Archiving)	Enables users to create a report containing the line item proposals and archive it with the PROCESS DIRECTOR document. In the parameters, you specify: ▪ the report to create, for example, `RFZALI20` (Payment list) ▪ the report variant to use ▪ the archiving document type to create (this must have the document class `PDF`) ▪ (optional) the format to use. This specifies, for example, the maximum number of lines and columns per page.

Process	Description
Parameters controlling document creation based on (Other)	Enables you to control which payment proposal data is retrieved from SAP. See *Configuring payment approval data retrieval* in the *PROCESS DIRECTOR SAP Configuration Guide* for more information.
Retrieve new documents from SAP on Worklist double-click (Other)	Turn on to retrieve new SAP payment proposals when users double-click a Payment Approval Worklist node. Payment proposals are also retrieved when the program `/EBY/PDPR_IMPORT` runs. Under certain circumstances this can lead to duplicate documents in PROCESS DIRECTOR. Turn this setting off if you are experiencing problems with duplicate documents.

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.
- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

Logical level	Package	Interface structure	Customer Includes
Header data	`/EBY/PDPR_MODEL`	`/EBY/PDPR_SHDR_IF`	`CI_EBY_PDPR_SHDR_DATA` `CI_EBY_PDPR_SHDR_DISP`

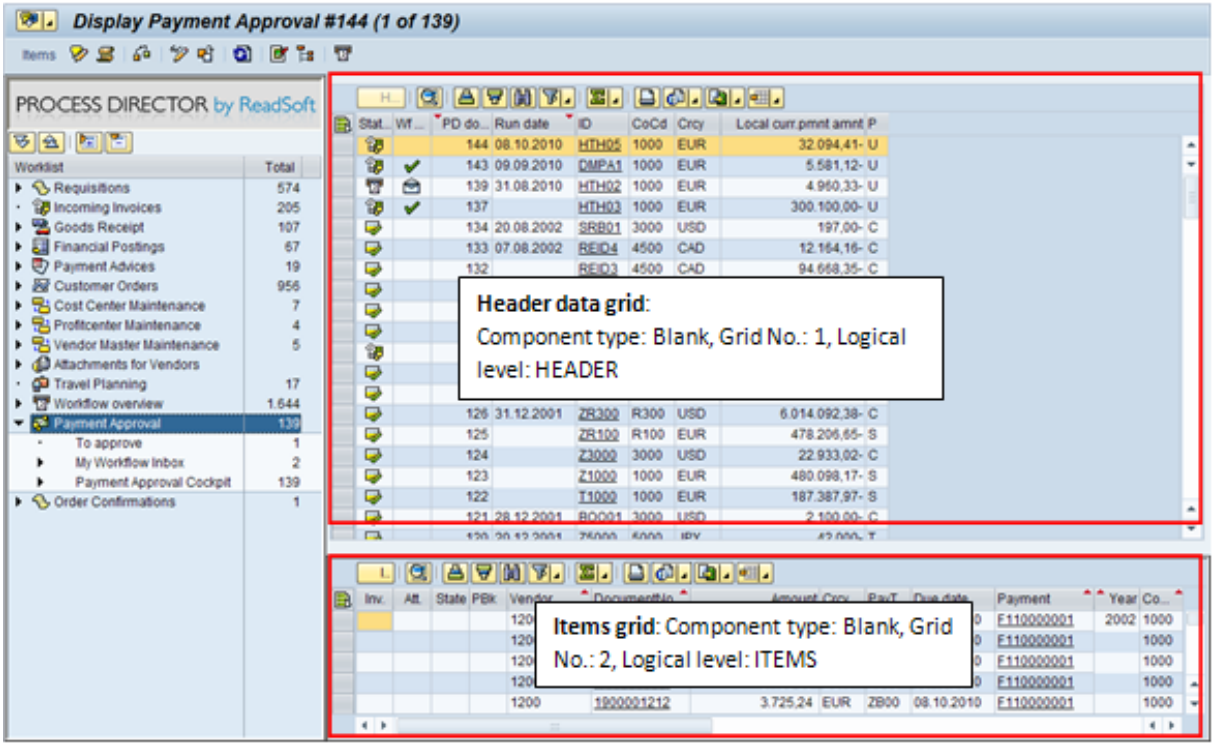
Logical level	Package	Interface structure	Customer Includes
Item data	/EBY/PDPR_MODEL	/EBY/PDPR_SITM_IF	CI_EBY_PDPR_SITM_DATA CI_EBY_PDPR_SITM_DISP

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Payment Approvals. See the *PROCESS DIRECTOR Configuration Guide* for more information.

By GUI element

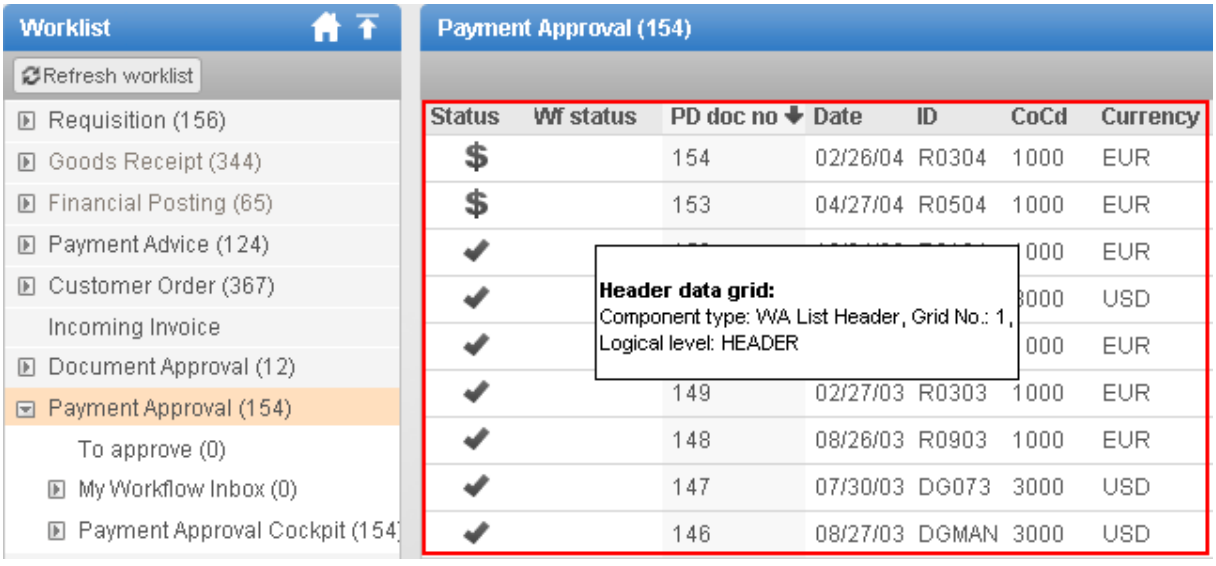
- SAP document overview list:



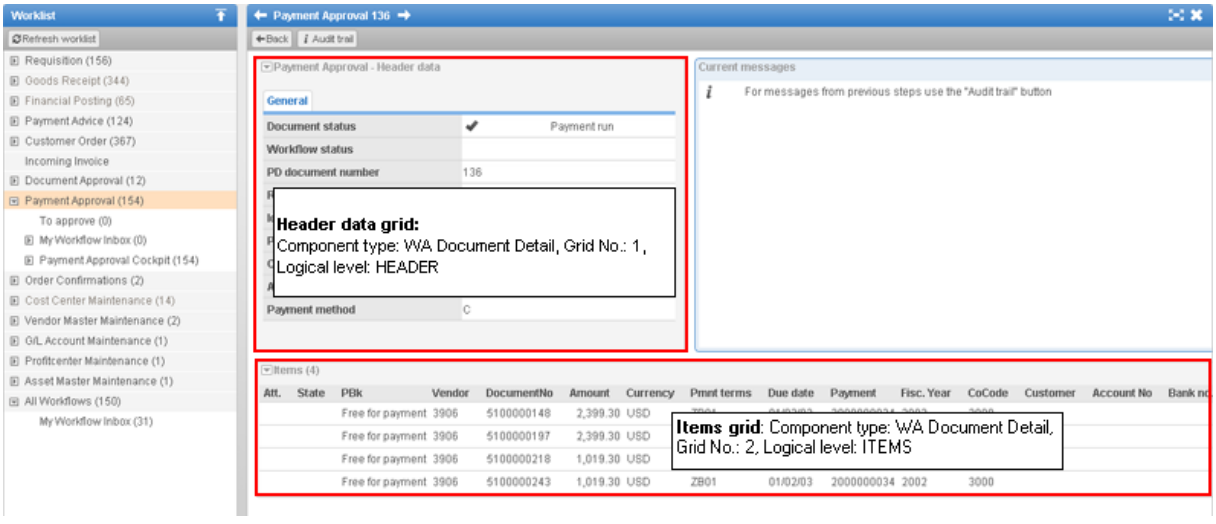
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

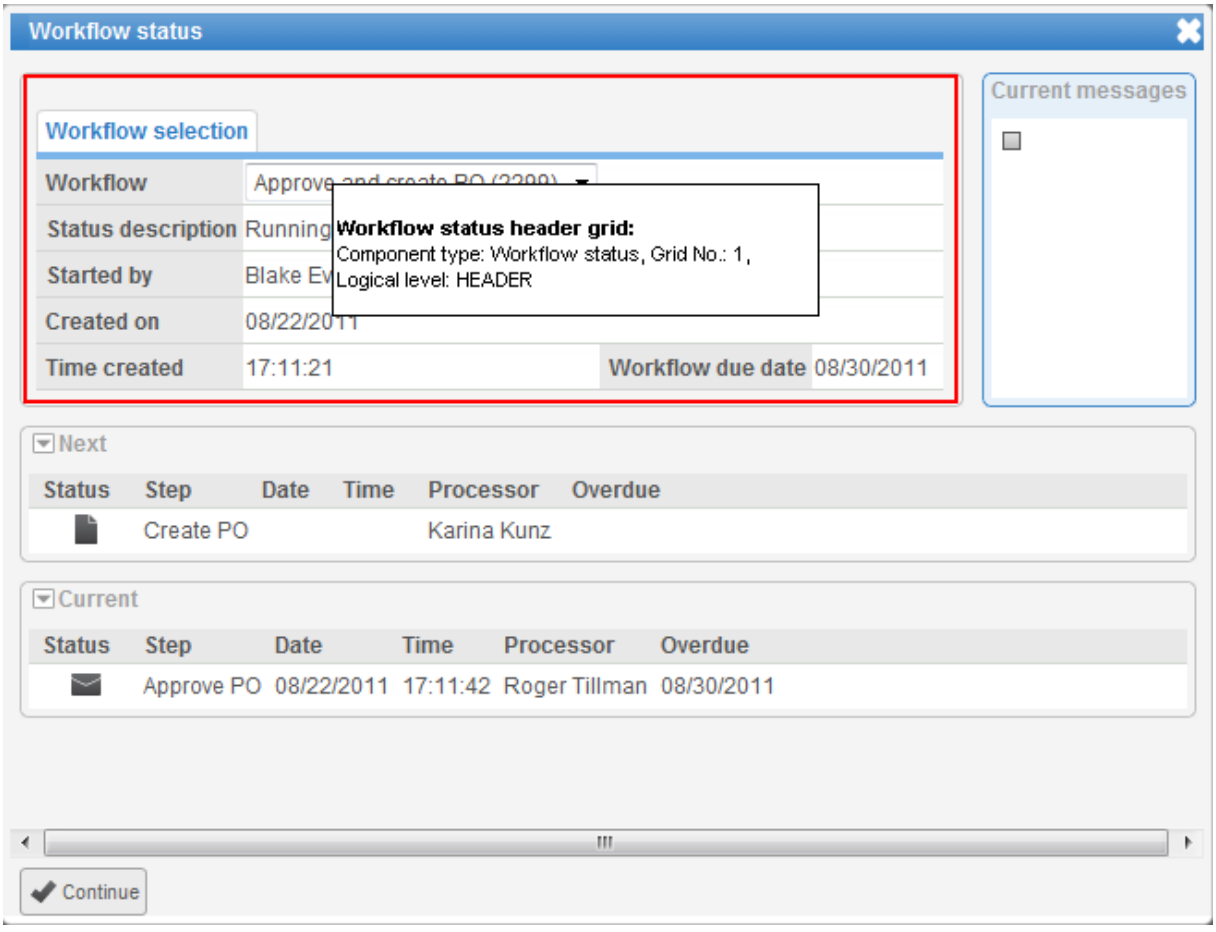
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
	1	HEADER	ZXY_PDPR_SGRID1 ZXY_PDPR_LGRID1	Header Header data
	2	ITEMS	ZXY_PDPR_SGRID2 ZXY_PDPR_LGRID2	Items Items
WA Document Detail	1	HEADER	ZXY_PDPR_SWA_DOC_GRID1 ZXY_PDPR_LWA_DOC_GRID1	Header Header data
WA Document Detail	2	ITEMS	ZXY_PDPR_SWA_DOC_GRID2 ZXY_PDPR_LWA_DOC_GRID2	Items Items

Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming proposals)	Btn. label Grid title (naming proposals)
WA List Header	1	HEADER	ZXY_PDPR_SWA_LIST_GRID1 ZXY_PDPR_LWA_LIST_GRID1	Header Header data
Workflow status	1	HEADER	ZXY_PDPR_SWA_STAT_GRID1 ZXY_PDPR_LWA_STAT_GRID1	Header Header data
Header data for archive log	1	HEADER	ZXY_PDPR_SARC_LOG_GRID1 ZXY_PDPR_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses Payment Approvals:

Status	Status Description
PP	Payment proposal
PR	Payment run

Text types

PROCESS DIRECTOR ships with this system text type for Payment Approvals:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X

Programs

The following programs are available for use with Payment Approvals:

- `/EBY/PDPR_IMPORT`

  Import payment proposals from SAP into PROCESS DIRECTOR.

- `/EBY/PDPR_INVOICE_UPDATER`

  This program is only relevant if you have payment proposal documents that were created in a version of PROCESS DIRECTOR prior to 7.1 Service Pack 1. This report reads the payment proposal line items and checks whether a corresponding PROCESS DIRECTOR Accounts Payable invoice document is available. If there is, the connection between the line item and the invoice is saved in the database.

/EBY/PDPR_IMPORT

Use the `/EBY/PDPR_IMPORT` program to import payment proposals from SAP into PROCESS DIRECTOR.

This program can be scheduled to run at regular intervals, for example, overnight. Proposals are still imported when the user clicks a payment approval Worklist node or the **Refresh** button, but only the proposals that have been created since the last program run are imported. This can considerably improve performance.

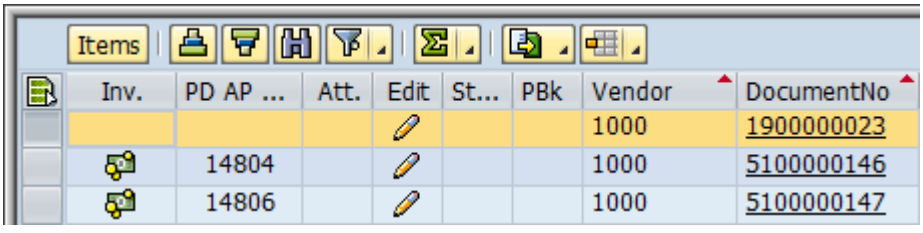
We recommend that you schedule this program if you have a large number of payment proposals, or payment proposals with many line items.

Note: If you are experiencing problems with duplicate PROCESS DIRECTOR documents created from SAP payment proposals, turn off the process parameter **Retrieve new documents from SAP on Worklist double-click**.

/EBY/PDPR_INVOICE_UPDATER

This program is only relevant if you have payment proposal documents that were created in a version of PROCESS DIRECTOR prior to 7.1 Service Pack 1. This report reads the payment proposal line items and checks whether a corresponding PROCESS DIRECTOR Accounts Payable invoice document is available. If there is, the connection between the line item and the invoice is saved in the database. This is necessary because prior to PROCESS DIRECTOR 7.1 SP1, these connections were read whenever a document was displayed, which had a negative effect on performance. From PROCESS DIRECTOR 7.1 SP1, these connections are automatically saved to the database.

A link to the PROCESS DIRECTOR Accounts Payable document is displayed in the **Inv.** column in the payment approval line item:



Requisitions (PO)

SAP Object type: `/EBY/PDPO`

Package: `/EBY/PDPO`

This chapter provides reference information for Requisitions.

Preset classes for Requisitions

The following preset classes are available for Requisitions in the PROCESS DIRECTOR standard. You can also use general preset classes that are available for all process types.

Preset accounting from previous item

`/EBY/CL_PDPO_PRESET_ACC`

Copies the values from the previous accounting item to new accounting items.

Preset company code from purchasing org

`/EBY/CL_PDPO_PRESET_COCO_PORG`

Sets the company code from the purchasing organization (if there is a unique assignment in table T024E).

Preset material master data

`/EBY/CL_PDPO_PRESET_MATMAS`

Sets the material group and the order unit values from the material master data (MARA table). If the order unit is not available in the master data, the base unit of measure is set instead.

Preset vendor data for header

`/EBY/CL_PDPO_PRESET_HDR_VENDOR`

Sets the **Currency**, **Incoterm 1**, **Incoterm 2** and **Payment terms** values from the vendor master data.

TEMPLATE: Preset PO header

`/EBY/CL_PDPO_PRESET_HEADER_TMP`

This preset class is provided as a template that you can use to configure your own custom presets for Requisitions header data.

Note: Do not add this preset class to the presets configuration. Instead, create your own preset class based on the template and add this custom preset class check to the configuration.

Checks for Requisitions

The following check events are available for Requisitions in the PROCESS DIRECTOR standard. You can also use general checks that are available for all process types.

Check Company Code

`/EBY/CL_PDPO_EVT_CHK_COMP_CODE`

Checks whether the company code entered in the requisition is maintained in SAP and is valid.

Check Vendors returned from OCI

`/EBY/CL_PDPO_EVT_CHK_OCI_VEND`

Checks if more than one vendor is returned from the OCI catalog.

Check SAP Purchase Order creation

`/EBY/CL_PDPO_EVT_CHK_POST`

Simulates the creation of the purchase order in SAP and returns all messages generated during the simulation.

Check purchase organization

`/EBY/CL_PDPO_EVT_CHK_PURCH_ORG`

Checks whether the company code entered in the requisition is maintained in SAP and is valid.

Check item quantity and account quantity sum or percentage

`/EBY/CL_PDPO_EVT_CHK_QUANTITY`

Checks the following and returns error messages if any apply:

- The item quantity differs from the account assignment quantity.
- The item quantity differs from the delivery schedule quantity.
- The total account assignment percentage distribution is not equal to 100%.

Check maintained schedule lines

`/EBY/CL_PDPO_EVT_CHK_SCHEDULE`

Checks if there are line items without a scheduled delivery date or with a delivery date in the past.

Check vendor

`/EBY/CL_PDPO_EVT_CHK_VENDOR`

Checks whether the vendor entered in the requisition is maintained in SAP for the given company code and purchasing organization, and whether the vendor is valid and not blocked.

Process parameters for Requisitions

The following process parameters are available for Requisitions. You can also use the general process parameters that are available for all process types.

Process	Description
Define XFlags for PO BAPI posting (Checks, Posting)	

Process	Description
Posting parameters (Posting)	Enables you to override the default settings for posting requisitions. **Parameter** / **Description** **Hold Purchase Order if faulty** **Hold Purchase Order if not faulty** **No message determination** **No messages necessary** **No authorization check** **No price transfer from last document** **Set update task local** **Call BAPI PO** **Wait on commit** **Call BAPI RQ** **Automatic source determination** **Skip faulty items**
Requisition specific posting parameters (Checks, Posting)	Determines whether a purchase order or a requisition will be created in SAP when a PROCESS DIRECTOR requisition document is posted.

Process	Description
Deactivation of popup to select contract (Other)	In the parameters, activate **Deactivation flag** to disable the popup that is displayed by default when a user creates a new requisition. This popup enables the user to create a new requisition based on an existing purchasing document or outline agreement. See also *Activating outline agreements* in the *PROCESS DIRECTOR SAP Configuration Guide*.

Determinations for Requisitions

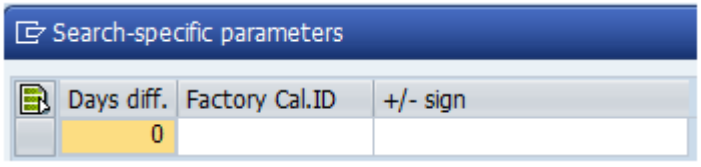
These determinations are available for Requisitions in the PROCESS DIRECTOR standard.

Purchase Order delivery date determination

`/EBY/CL_PDPO_DET_SCHED_INIT`

Determines the delivery date based on a specific number of working days after the document date of the requisition. This calculated delivery date is entered for all line items of the requisition.

The number of working days can be entered in the parameters of the **Purchase order delivery date determination by working days** search (`/EBY/CL_PDPO_DET_SCHED_WDAYS`). If you also enter a factory calendar ID, the delivery date calculation takes this calendar into account.

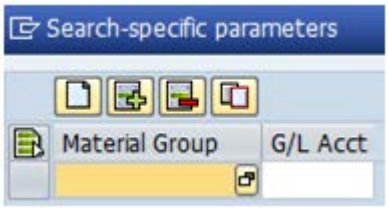


Purchase Order G/L account determination

`/EBY/CL_PDPO_DET_GLACCT_INIT`

Determines the G/L account based on the material group entered in the requisition line item.

Material groups and their corresponding G/L account numbers can be specified in the search parameters of the **Purchase order G/L account determination by mapping matgrp** search (`/EBY/CL_PDPO_DET_GLACCT_MATGRP`).

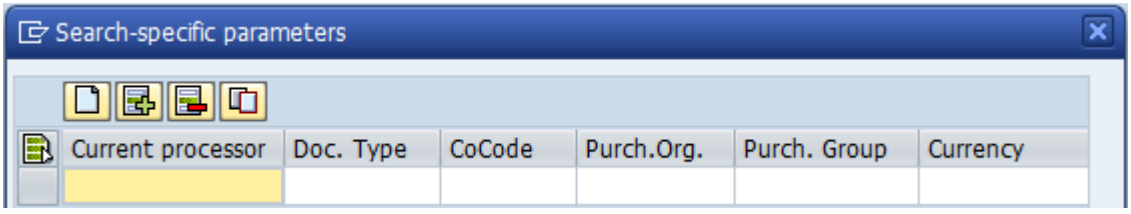


Purchase Order org. data determination

`/EBY/CL_PDPO_DET_ORG_INIT`

Determines the organization data based on the current processor of the document.

Processors and their corresponding organization data can be entered in the parameters of the **Purchase order org. data determination by mapping table** search.



Purchase Order vendor determination from OCI

`/EBY/CL_PDPO_DET_VENDOR_INIT`

Determines the vendor from the OCI catalog based on (depending on the defined searches):

Search type	Search name
The account number	`/EBY/CL_PDPO_DET_VENDOR_ACC`
The account number from the purchase order	`/EBY/CL_PDPO_DET_VENDOR_ACCPO`
A mapping table	`/EBY/CL_PDPO_DET_VENDOR_MAP`

Document model customization

The document model provides an interface structure for each logical level containing all fields that are available for that level. Using the customer includes (CIs) provided in this structure, you can extend the document model to include customer-specific fields. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

- Use the include ending in _DATA if the custom field should be automatically persisted in the database.

- Use the include ending in _DISP if you only want to automatically calculate a field at runtime and display it.

Important: Never add customer-specific fields directly to the customer include, but use your own customer-specific structure instead. For example, first create a customer-specific structure corresponding to each used CI, include that structure in the CI and put all customer-specific fields in the customer-specific structure. **Do not use an APPEND structure.** Set the enhancement category of the customer-specific structure to **Can Be Enhanced, Character-Type or Numeric-Type** (menu **Extras**). Never use the enhancement category **Can Be Enhanced (Deep)**. See the SAP documentation for more information.

Logical level	Package	Interface structure	Customer Includes
Header data	/EBY/PDPO_MODEL	/EBY/PDPO_SHDR_IF	CI_EBY_PDPO_SHDR_DATA CI_EBY_PDPO_SHDR_DISP
Item data	/EBY/PDPO_MODEL	/EBY/PDPO_SITM_IF	CI_EBY_PDPO_SITM_DATA CI_EBY_PDPO_SITM_DISP
Account data	/EBY/PDPO_MODEL	/EBY/PDPO_SACC_IF	CI_EBY_PDPO_SACC_DATA CI_EBY_PDPO_SACC_DISP
Item conditions data	/EBY/PDPO_MODEL	/EBY/PDPO_SICO_IF	CI_EBY_PDPO_SICO_DATA CI_EBY_PDPO_SICO_DISP
Partner data	/EBY/PDPO_MODEL	/EBY/PDPO_SPAR_IF	CI_EBY_PDPO_SPAR_DATA CI_EBY_PDPO_SPAR_DISP
Schedule data	/EBY/PDPO_MODEL	/EBY/PDPO_SSCH_IF	CI_EBY_PDPO_SSCH_DATA CI_EBY_PDPO_SSCH_DISP
Header texts	/EBY/PDPO_MODEL	/EBY/PDPO_SHTX_IF	CI_EBY_PDPO_SHTX_DATA CI_EBY_PDPO_SHTX_DISP
Item texts	/EBY/PDPO_MODEL	/EBY/PDPO_SITX_IF	CI_EBY_PDPO_SITX_DATA CI_EBY_PDPO_SITX_DISP

BAPI structure fields

Some document data models include BAPI structures. These structures can contain a large number of fields, many of which are not used. Therefore, most document data model BAPI structure fields are transient by default.

However, a customer may require that one or more of these fields be made persistent. PROCESS DIRECTOR offers an easy way to fulfill this requirement. In SAP transaction `SE80`, simply add the BAPI structure field(s) to the appropriate customer include.

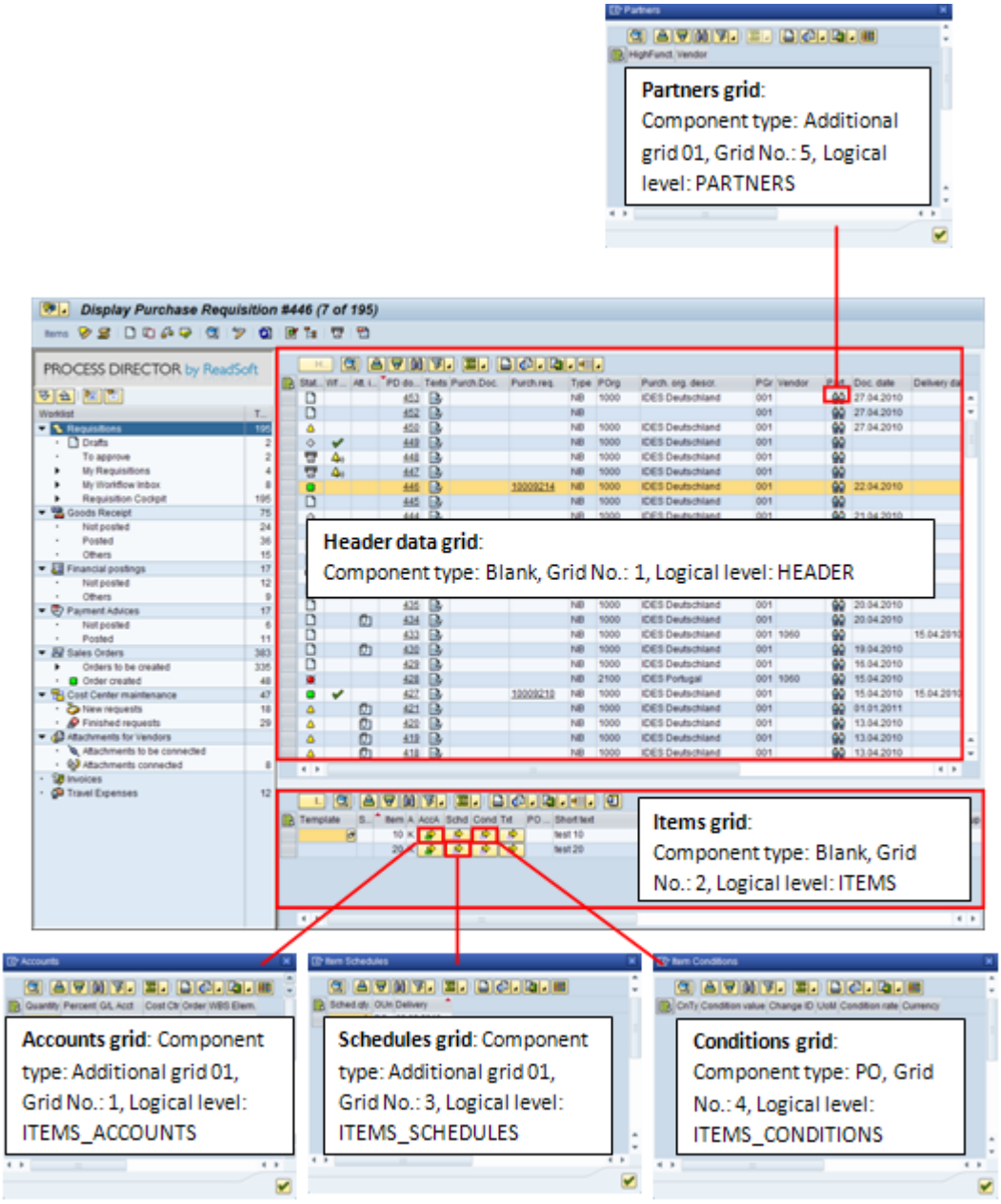
Logical level	Package	Structure	Customer Include
Header data	`/EBY/PDPO_PERSISTENCE`	`/EBY/PDPO_THDR`	`CI_EBY_PDPO_SHDR_BAPI`
Item data	`/EBY/PDPO_PERSISTENCE`	`/EBY/PDPO_TITM`	`CI_EBY_PDPO_SITM_BAPI`
Account data	`/EBY/PDPO_PERSISTENCE`	`/EBY/PDPO_TACC`	`CI_EBY_PDPO_SACC_BAPI`
Item conditions data	`/EBY/PDPO_PERSISTENCE`	`/EBY/PDPO_TICO`	`CI_EBY_PDPO_SICO_BAPI`
Partner data	`/EBY/PDPO_PERSISTENCE`	`/EBY/PDPO_TPAR`	`CI_EBY_PDPO_SPAR_BAPI`
Schedule data	`/EBY/PDPO_PERSISTENCE`	`/EBY/PDPO_TSCH`	`CI_EBY_PDPO_SSCH_BAPI`
Header texts	`/EBY/PDPO_PERSISTENCE`	`/EBY/PDPO_THTX`	`CI_EBY_PDPO_SHTX_BAPI`
Item texts	`/EBY/PDPO_PERSISTENCE`	`/EBY/PDPO_TITX`	`CI_EBY_PDPO_SITX_BAPI`

View model customization

This section provides detailed information about which settings to enter when creating a customer view model for Requisitions. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

By GUI element

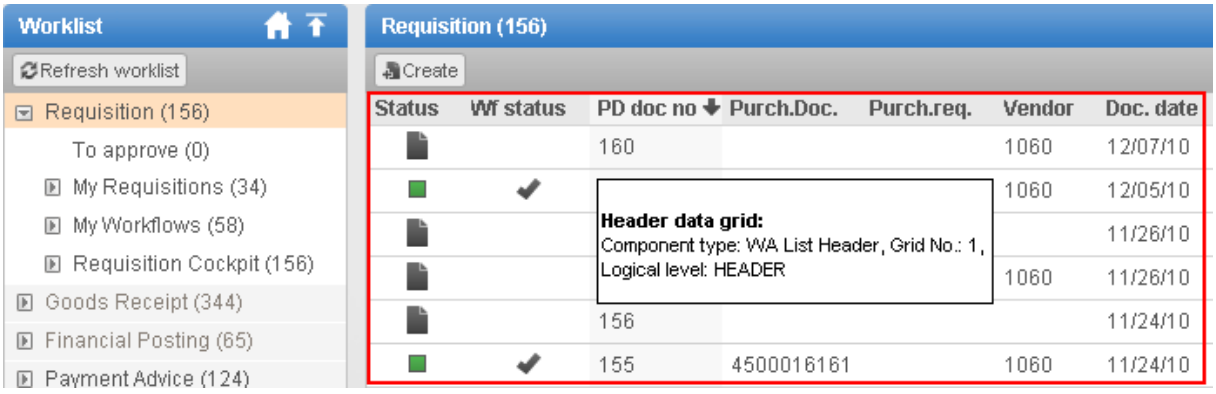
- SAP document overview list:



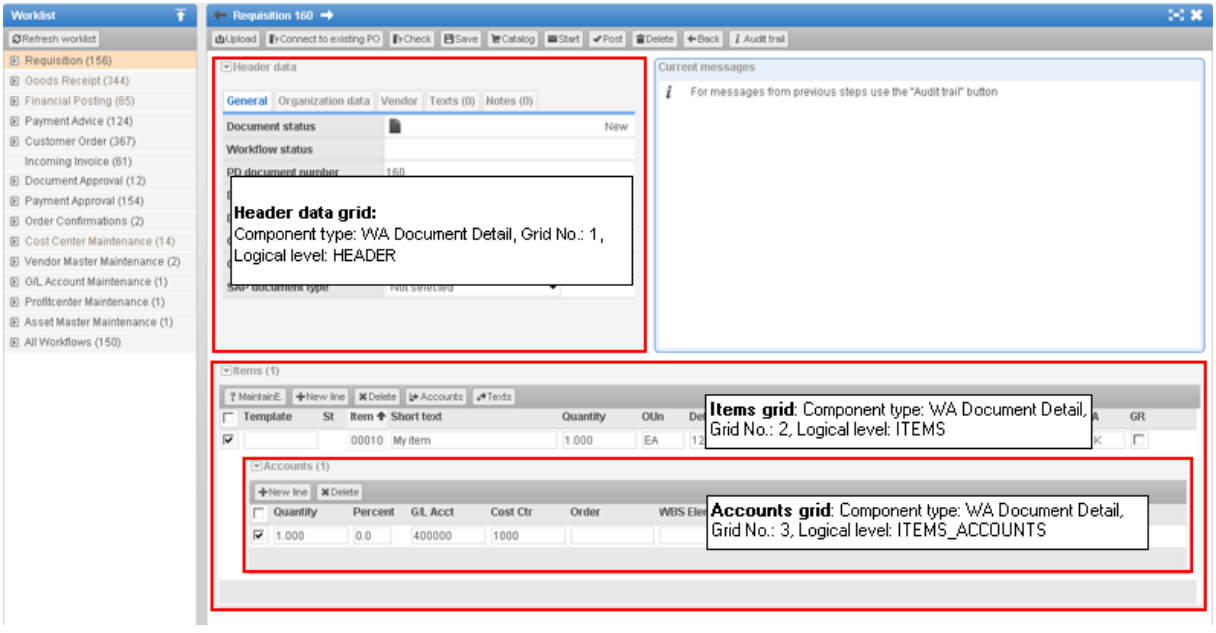
- SAP document detail:

To maintain the SAP GUI detail screen, use SAP transaction `/EBY/PDVI_VDSTC`.

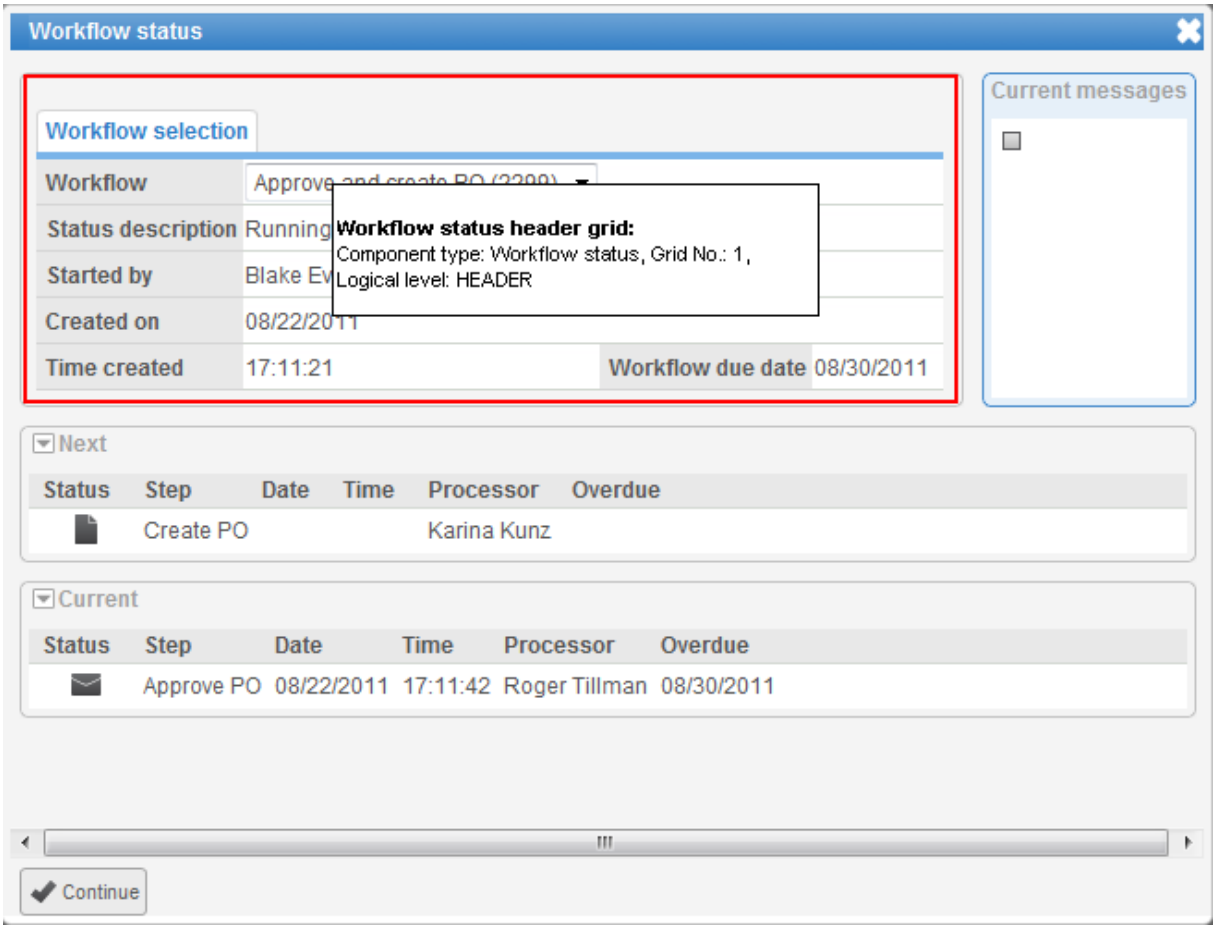
- Web Application document overview list:



- Web Application document detail:



- Workflow status:



By view model component type

GUI area	Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming suggestions)	Btn. label Grid title (naming suggestions)
SAP document list		1	HEADER	ZXY_PDPO_SGRID1 ZXY_PDPO_LGRID1	Header Header data
SAP Line items grid		2	ITEMS	ZXY_PDPO_SGRID2 ZXY_PDPO_LGRID2	Items Items
SAP Accounts grid	Additional grid 01	1	ITEM_ACCOUNTS	ZXY_PDPO_SGRID3 ZXY_PDPO_LGRID3	Accounts Accounts

GUI area	Comp. type	Grid No.	Logical level	Cust. display struct. Cust. table type name (naming suggestions)	Btn. label Grid title (naming suggestions)
SAP Schedules grid	Additional grid 01	3	ITEM_SCHEDULES	ZXY_PDPO_SGRID4 ZXY_PDPO_LGRID4	Schedules Item Schedules
SAP Conditions grid	Additional grid 01	4	ITEM_CONDITIONS	ZXY_PDPO_SGRID5 ZXY_PDPO_LGRID5	Conditions Item Conditions
SAP Partners grid	Additional grid 01	5	PARTNERS	ZXY_PDPO_SGRID7 ZXY_PDPO_LGRID7	Partners Partners
Web Application document detail header data	WA Document Detail	1	HEADER	ZXY_PDPO_SWA_DOC_GRID1 ZXY_PDPO_LWA_DOC_GRID1	Header Header data
Web Application document detail Line items grid	WA Document Detail	2	ITEMS	ZXY_PDPO_SWA_DOC_GRID2 ZXY_PDPO_LWA_DOC_GRID2	Items Items
Web Application Accounts grid	WA Document Detail	3	ITEM_ACCOUNTS	ZXY_PDPO_SWA_DOC_GRID3 ZXY_PDPO_LWA_DOC_GRID3	Accounts Accounts
Web Application document list	WA List Header	1	HEADER	ZXY_PDPO_SWA_LIST_GRID1 ZXY_PDPO_LWA_LIST_GRID1	Header Header data
Workflow status popup	Workflow status	1	HEADER	ZXY_PDPO_SWA_STAT_GRID1 XY_PDPO_LWC_STAT_GRID1	Header Header data
Archive log	Header data for archive log	1	HEADER	ZXY_PDPO_SARC_LOG_GRID1 ZXY_PDPO_LARC_LOG_GRID1	Header Header data

Document statuses

PROCESS DIRECTOR ships with these system document statuses/substatuses for Requisitions:

Statuses

Status	Status Description
ER	Not postable
NP	Postable
NW	New
OK	Posted
WR	Postable with warning(s)

Substatuses

Substatus	Description
PM	Purchase Order Multiple
PO	Purchase Order
RM	Purchase Requisition Multiple
RQ	Purchase Requisition

Text types

PROCESS DIRECTOR ships with these system text types for Requisitions:

Logical level	Text object	Description	ID	Description	Multiple
HEADER	/EBY/PD	PROCESS DIRECTOR texts	PDN	PROCESS DIRECTOR note	X
HEADER	EKKO	Purchasing doc. header texts	F01	Header text	
ITEMS	EBAN	Purchase requisition texts	B04	Material PO text	
ITEMS	EKPO	Purchasing doc. item texts	F03	Material PO text	

User exits / BAdIs

PROCESS DIRECTOR provides the following user exits/BAdIs for use with Requisitions:

Initialize values

Posting

Check

BAdI Initialize values (Requisitions)

This user exit / BAdI currently provides the interface method `INITIALIZE_VALUES` that you can implement to add your own customer rules for initializing field values for Requisitions. You can use it as an alternative or supplement to presets.

INITIALIZE_VALUES method

The `INITIALIZE_VALUES` method offers access to the Requisition document model logical level structures as `CHANGING` parameters. To initialize one or more fields, simply assign values to them in your coding, like this:

```
METHOD /eby/if_pdpo_badi_init_val~initialize_values .
```

```
  cs_po_hdr-currency = 'GBP'.
ENDMETHOD.
```

Calling sequence

In the PROCESS DIRECTOR standard, this user exit is called from the final event of the `CREAT1 Create document` action. It runs *after* presets are applied, but *before* the document is actually created on the database.

Important: Document creation will be prevented if an `INITIALIZE_VALUES` method implementation issues an error message, unless the error message is filtered with a `Remove message` filter.

Method signature

Since the user exit / BAdI is document-type-specific, the `INITIALIZE_VALUES` method signature depends on the document type.

Type	Parameter	Description
	`IR_OBJECT`	Requisitions object
	`CS_BO_HDR`	Data fields of a PD base object
	`CS_PO_HDR`	Structure containing requisition header data
	`CT_ITEMS`	Table containing requisition line item data
	`CT_ITEM_ACCOUNTS`	Table containing requisition accounts data
	`CT_ITEM_SCHEDULES`	Table containing requisition schedules data
	`CT_ITEM_CONDITIONS`	Table containing requisition conditions data
	`CT_PARTNERS`	Table containing partners

BAdI Posting (Requisitions)

This user exit / BAdI currently provides the interface methods `BEFORE_POST`, `BEFORE_POST_RQ`, `AFTER_POST` and `AFTER_POST_RQ` that you can implement to add your own customer logic before or after a Requisition document is posted.

Both interface methods offer access to SAP PO BAPI structures as parameters:

- The `BEFORE_POST` interface method exposes the PO BAPI structures as `CHANGING` parameters pre-populated with the PO data to be posted to SAP.
- THE `AFTER_POST` interface method exposes the PO BAPI structures as `IMPORTING` parameters containing the PO data that has just been posted to SAP.

BEFORE_POST method

Calling sequence

This method is called from the action `POST_DOC Post document`. It runs immediately *before* document posting occurs.

Method signature

Type	Parameter	Description
	`IR_OBJECT`	Requisitions object
	`IS_BO_HDR`	Data fields of a PD base object
	`IS_PO_HDR`	Requisition header data
	`IT_ITEMS`	Requisition line item data
	`IT_ITEM_ACCOUNTS`	Requisition accounts data
	`IT_ITEM_SCHEDULES`	Requisition schedules data
	`IT_ITEM_CONDITIONS`	Requisition conditions data
	`IT_PARTNERS`	Requisition partner data

Type	Parameter	Description
	CS_HEADER	Requisition header data
	CS_HEADERX	Requisition header data (change toolbar)
	CS_ADDRVENDOR	Requisition header: address structure BAPIADDR1 for vendor
	CS_EXPHEADER	Requisition header data
	CT_ITEMS	Table type for BAPIMEPOITEM
	CT_ITEMSX	Table type for BAPIMEPOITEMX
	CT_ACCOUNTS	Table type for BAPIMEPOACCOUNT
	CT_ACCOUNTSX	Table type for BAPIMEPOACCOUNTX
	CT_ACCPROFITSEGMENTS	Table type for BAPIMEPOACCOUNTPROFITSEGMENT
	CT_SCHEDULES	Table Type for BAPIMEPOSCHEDULE
	CT_SCHEDULESX	Table Type for BAPIMEPOSCHEDULX
	CT_CONDHEADER	Table Type for BAPIMEPOCONDHEADER
	CT_CONDHEADERX	Table Type for BAPIMEPOCONDHEADERX
	CT_CONDITIONS	Table Type for BAPIMEPOCOND
	CT_CONDITIONSX	Table Type for BAPIMEPOCONDX
	CT_LIMITS	Table Type for BAPIESUHC
	CT_CONTRACTLIMITS	Table Type for BAPIESUCC

Type	Parameter	Description
	CT_SERVICES	Table Type for BAPIESLLC
	CT_SRVACCESSVALUES	Table Type for BAPIESKLC
	CT_PARTNERS	Table Type for BAPIEKKOP
	CT_ADDRDELIVERY	Table Type for BAPIMEPOADDRDELIVERY
	CT_HEADERTEXTS	Table Type for BAPIMEPOTEXTHEADER
	CT_ITEMTEXTS	Table Type for BAPIMEPOTEXT
	CT_SERVICESTEXT	Table Type for BAPIESLLTX
	CT_EXTENSIONIN	Table with BAPIPAREX as Row Type
	CT_EXTENSIONOUT	Table with BAPIPAREX as Row Type
	CT_RETURN	Return parameter table
	CT_ITEM_HIST_TOTALS	Table of structure BAPIEKBES
	CT_SELECTED_ITEMS	Select options for GUIDs
	CT_ITEM_MAPPING	Item numbers mapping

BEFORE_POST_RQ method

Method signature

Type	Parameter	Description
	IR_OBJECT	Requisitions object

Type	Parameter	Description
	IS_BO_HDR	Data fields of a PD base object
	IS_PO_HDR	Requisition header data
	IT_ITEMS	Requisition line item data
	IT_ITEM_ACCOUNTS	Requisition accounts data
	IT_ITEM_SCHEDULES	Requisition schedules data
	IT_ITEM_CONDITIONS	Requisition conditions data
	IT_PARTNERS	Requisition partner data
	CC_VENDOR	Account number of the vendor
	CT_SERVICES	Table Type for BAPIESLLC
	CT_SERVICES_TEXTS	Table Type for BAPIESLLTX
	CT_SRV_ACCASS_VALUES	Table Type for BAPIESKLC
	CT_ACCOUNTS	Table Type for BAPIEBKN
	CT_ACCOUNTS_OLD	Table Type for BAPIEBKN
	CT_ADDRDELIVERY	Type BAPIMERQADDRDELIVERY
	CT_CONTRACT_LIMITS	Table Type for BAPIESUCC
	CT_EXTENSIONIN	Table Type for BAPIPAREX
	CT_ITEMS	Table Type for BAPIEBAN

Type	Parameter	Description
	CT_ITEMS_OLD	Table Type for BAPIEBANC
	CT_ITEM_MAPPING	Item numbers mapping
	CT_ITEM_TEXTS	Table Type for BAPIEBANTX
	CT_ITEM_TEXTS_OLD	Table Type for BAPIEBANTX
	CT_LIMITS	Table Type for BAPIESUHC
	CT_RETURN	Table of structure BAPIRETURN
	CT_SELECTED_ITEMS	Select options for GUIDs

AFTER_POST method

Calling sequence

This method is called from the action `POST_SUCCESS Posting succeeded`. It runs after the document status `Posted` has been set.

Note: If document posting fails, `AFTER_POST` is *not* called.

Method signature

Type	Parameter	Description
	IR_OBJECT	Requisitions object
	IS_HEADER	Requisition header data
	IS_HEADERX	Requisition header data (change toolbar)
	IS_ADDRVENDOR	PO Header: Address Structure BAPIADDR1 for Vendor
	IS_EXPHEADER	Purchase Order Header Data

Type	Parameter	Description
	IT_ITEMS	Table Type for BAPIMEPOITEM
	IT_ITEMSX	Table Type for BAPIMEPOITEMX
	IT_ACCOUNTS	Table Type for BAPIMEPOACCOUNT
	IT_ACCOUNTSX	Table Type for BAPIMEPOACCOUNTX
	IT_ACCPROFITSEGMENTS	Table Type for BAPIMEPOACCOUNTPROFITSEGMENT
	IT_SCHEDULES	Table Type for BAPIMEPOSCHEDULE
	IT_SCHEDULESX	Table Type for BAPIMEPOSCHEDULX
	IT_CONDHEADER	Table Type for BAPIMEPOCONDHEADER
	IT_CONDHEADERX	Table Type for BAPIMEPOCONDHEADERX
	IT_CONDITIONS	Table Type for BAPIMEPOCOND
	IT_CONDITONSX	Table Type for BAPIMEPOCONDX
	IT_LIMITS	Table Type for BAPIESUHC
	IT_CONTRACTLIMITS	Table Type for BAPIESUCC
	IT_SERVICES	Table Type for BAPIESLLC
	IT_SRVACCESSVALUES	Table Type for BAPIESKLC
	IT_PARTNERS	Table Type for BAPIEKKOP
	IT_ADDRDELIVERY	Table Type for BAPIMEPOADDRDELIVERY

Type	Parameter	Description
	IT_HEADERTEXTS	Table Type for BAPIMEPOTEXTHEADER
	IT_ITEMTEXTS	Table Type for BAPIMEPOTEXT
	IT_SERVICESTEXT	Table Type for BAPIESLLTX
	IT_EXTENSIONIN	Table with BAPIPAREX as Row Type
	IT_EXTENSIONOUT	Table with BAPIPAREX as Row Type
	IT_RETURN	Return parameter table
	IT_ITEM_HIST_TOTALS	Table of structure BAPIEKBES
	IT_SELECTED_ITEMS	Select options for GUIDs
	IT_ITEM_MAPPING	Item numbers mapping
	CS_BO_HDR	Data fields of a PD base object
	CS_PO_HDR	Requisition header data
	CT_ITEMS	Requisition line item data
	CT_ITEM_ACCOUNTS	Requisition accounts data
	CT_ITEM_SCHEDULES	Requisition schedules data
	CT_ITEM_CONDITIONS	Requisition conditions data
	CT_PARTNERS	Requisition partner data

AFTER_POST_RQ method

Method signature

Type	Parameter	Description
	IR_OBJECT	Requisitions object
	IC_VENDOR	Account number of the vendor
	IT_SERVICES	Table Type for BAPIESLLC
	IT_SERVICES_TEXTS	Table Type for BAPIESLLTX
	IT_SRV_ACCASS_VALUES	Table Type for BAPIESKLC
	IT_ACCOUNTS	Table Type for BAPIEBKN
	IT_ACCOUNTS_OLD	Table Type for BAPIEBKN
	IT_ADDRDELIVERY	Type BAPIMERQADDRDELIVERY
	IT_CONTRACT_LIMITS	Table Type for BAPIESUCC
	IT_EXTENSIONIN	Table Type for BAPIPAREX
	IT_ITEMS	Table Type for BAPIEBAN
	IT_ITEMS_OLD	Table Type for BAPIEBANC
	IT_ITEM_MAPPING	Item numbers mapping
	IT_ITEM_TEXTS	Table Type for BAPIEBANTX
	IT_ITEM_TEXTS_OLD	Table Type for BAPIEBANTX
	IT_LIMITS	Table Type for BAPIESUHC

Type	Parameter	Description
	IT_RETURN	Table of structure BAPIRETURN
	IT_SELECTED_ITEMS	Select options for GUIDs
	CS_BO_HDR	Data fields of a PD base object
	CS_PO_HDR	Requisition header data
	CT_ITEMS	Requisition line item data
	CT_ITEM_ACCOUNTS	Requisition accounts data
	CT_ITEM_SCHEDULES	Requisition schedules data
	CT_ITEM_CONDITIONS	Requisition conditions data
	CT_PARTNERS	Requisition partner data

BAdI Check (Requisitions)

This user exit / BAdI currently provides the interface method `CHECK` that you can implement to add your own customer check routine for Requisitions. You can use it as an alternative or supplement to checks.

CHECK

The `CHECK` method offers access to the Requisitions document model logical level structures as `IMPORTING` parameters.

Here is a sample implementation of the `CHECK` interface method that shows how to generate a message from a BAPI call. Note that you must export the proper message group as a parameter to the `ADD_BAPIMSG` method call:

```
METHOD /EBY/IF_PDPO_BADI_CHECK~CHECK.
```

```
  DATA: ls_bapireturn TYPE bapireturn,
        lr_bapi TYPE REF TO /eby/cl_bapi.
  CREATE OBJECT lr_bapi.
  ls_bapireturn = lr_bapi->check_purch_org_exists(
 is_po_hdr-purch_org ).
  IF NOT ls_bapireturn IS INITIAL.
    CALL METHOD /eby/cl_pdbo_exc=>add_bapimsg
      EXPORTING
        ic_grp  = /eby/if_pdbo_con_exc=>grp_check
        is_bapi = ls_bapireturn.
  ENDIF.
ENDMETHOD.
```

Method signature

Type	Parameter	Description
	IR_OBJECT	Requisitions object
	IS_BO_HDR	Data fields of a PD base object
	IS_PO_HDR	Requisition header data
	IT_ITEMS	Requisition line item data
	IT_ITEM_ACCOUNTS	Requisition accounts data
	IT_ITEM_SCHEDULES	Requisition schedules data
	IT_ITEM_CONDITIONS	Requisition conditions data
	IT_PARTNERS	Requisition partner data

Calling sequence

The `CHECK` BAdI method is called from the actions `CHECK Check document` and `CHECKM Check documents (multiple)`.

It is run *after* activated determination events, activated user exit / BAdI **Determination** implementations, and activated check events are executed, but immediately *before* the document status is set depending on the check result.

Like standard PROCESS DIRECTOR check events, a user exit / BAdI **Check** implementation communicates its results by issuing messages.

Index

A

B

C

D

E

F

G

H

I

M

O

P

Q

R

U

V

W