ReadSoft PROCESS DIRECTOR

Reference Guide

Version: 7.6

Date: 2020-05-06

Table of Contents

About this guide

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

The guide provides reference information for ReadSoft PROCESS DIRECTOR.

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

- *PROCESS DIRECTOR Import Guide*
- *PROCESS DIRECTOR Configuration Guide*
- *PROCESS DIRECTOR Advanced Configuration Guide*
- *PROCESS DIRECTOR Troubleshooting Guide*

General reference

Available query classes

You use query classes in the ranges configuration of semi-dynamic and dynamic Worklist nodes to supply parts of the query at runtime. For example, the query classes */EBY/CL_PDBO_WL_QRY_PD_USER_TYPE* and */EBY/CL_PDBO_WL_QRY_PD_USER* determine the name of the currently logged on user and display all documents assigned to that user in the Worklist node.

The following standard query classes are available.

Alpha conversion

/EBY/CL_PDBO_WL_QRY_ALPHA_CONV

Converts 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 are 0000001000, 0000002000, and so on. If you use the alpha conversion query, the node names are 1000, 2000, and so on.

Current day (sy-datum) +/- days

/EBY/CL_PDBO_WL_QRY_CUR_DAY

Includes documents whose date is within a specific time frame, such as documents from the last week or the last month.

Add the date field to the ranges configuration and in the **Range details** dialog box, select the query class. Close and reopen the dialog box and type the number of days in the **Days** field.

Time stamp of the current day +/- days

/EBY/CL_PDBO_WL_QRY_CUR_DAY_TS

Includes documents whose time stamp is within a specific time frame.

Add the time stamp field to the ranges configuration in the **Range details** dialog box, select the query class. Close and reopen the dialog box and type the number of days in the **Days** field.

Active substitutes query

/EBY/CL_PDBO_WL_QRY_PD_SUB_ACT

Includes documents of active and passive substitutes.

When you use this query class, the node's view must include the **WC_PROCESSER** and **WC_PROC_TYPE** fields.

PD user substitutions

/EBY/CL_PDBO_WL_QRY_PD_SUBST

Include documents of active substitutes only.

When you use this query class, the node's view must include the **WC_PROCESSER** and **WC_PROC_TYPE** fields.

PD user name

/EBY/CL_PDBO_WL_QRY_PD_USER

Include documents of the currently logged on user. To uniquely identify the user, you must also add the */EBY/CL_PDBO_WL_QRY_PD_USER_TY* query class.

PD user type

/EBY/CL_PDBO_WL_QRY_PD_USER_TY

You can use this query class to include documents of the currently logged on user type. To uniquely identify the user, you must also add the */EBY/CL_PDBO_WL_QRY_PD_USER* query class.

State description

/EBY/CL_PDBO_WL_QRY_STATE_DESC

Creates dynamic nodes for individual document statuses and displays status descriptions such as **New** or **Processed** instead of the corresponding status value **NN** or **OK**.

Available mapping conversions

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

Add parameter to value

/EBY/PDBO_MAF_ADD

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.

Write input to a text defined by text ID

/EBY/PDBO_MAF_SET_TEXTID

Specifies which long texts are transferred from EDI data to PROCESS DIRECTOR. This mapping conversion simply writes the specified text ID to memory. To transfer the long texts, you must also create a mapping conversion using the */EBY/PDBO_MAF_ADD_TEXT* function.

1. Create a mapping conversion using the */EBY/PDBO_MAF_SET_TEXTID* function. In the parameters, specify the text object. Optionally, specify a text ID. If you do not specify a text ID, the current text ID from EDI is written to memory.
2. In the EDI data mapping, map the *E1EDKT1* text identification segment to the external field name *TDID* and add the mapping conversion in the **Mapping** field.
3. Create a mapping conversion using the */EBY/PDBO_MAF_ADD_TEXT* function. Specify the same text object as in the */EBY/PDBO_MAF_SET_TEXTID* mapping conversion. Optionally, specify a text ID.

Write input to a text defined by text ID

/EBY/PDBO_MAF_ADD_TEXT

Saves an external string to a PROCESS DIRECTOR text. You can also use this mapping conversion together with */EBY/PDBO_MAF_SET_TEXTID* to transfer long texts from EDI data.

Example:

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.

Process arithmetic functions on input

/EBY/PDBO_MAF_ARITHMETIC

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.

Assign a variable to output

/EBY/PDBO_MAF_ASSIGN

*A*ssigns 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*.

Initialize the value

/EBY/PDBO_MAF_CLEAR

*R*emoves 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.

Adjust date (add/subtract working days)

/EBY/PDBO_MAF_DATEADJUST

*C*hanges the date by adding or subtracting the number of working days specified in the parameters.

Check/convert date to internal format

/EBY/PDBO_MAF_DATECONVERT

*C*hecks 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).

Delete first occurrence of parameter from output

/EBY/PDBO_MAF_DEL1ST

*D*eletes 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.

Delete all occurrences of a parameter from input

/EBY/PDBO_MAF_DELETE

Replaces all occurrences of the string specified in the parameters.

Divide input with parameter

/EBY/PDBO_MAF_DIVIDE

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.

Remove all characters which do not appear in parameter list

/EBY/PDBO_MAF_FILTER_ALLOWED

*R*emoves 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.

Use fixed value from parameters

/EBY/PDBO_MAF_FIXEDVALUE

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.

Skip next function unless condition is met

/EBY/PDBO_MAF_IF

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.

Skip next function unless condition is met

/EBY/PDBO_MAF_IF_LEN

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.

Change the value to lower case

/EBY/PDBO_MAF_LOWERCASE

Changes the value of the field to lower case characters.

Example: Changes the field value *TEXT* or *Text* to *text*.

Multiply input with parameter

/EBY/PDBO_MAF_MULTIPLY

*F*or 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.

Change the value by removing all space characters

/EBY/PDBO_MAF_NOSPACE

Removes all spaces from the field value.

Example: Converts the telephone number 001 555 21 21 to 0015552121.

Map different field from the same level

/EBY/PDBO_MAF_OTHER_FIELD

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.

Append parameter to value

/EBY/PDBO_MAF_POSTFIX

Adds the string specified in the parameters at the end of the field value.

Prepend parameter to value

/EBY/PDBO_MAF_PREFIX

Adds the string specified in the parameters at the beginning of the field value.

Replace all occurrences of parameter A with B

/EBY/PDBO_MAF_REPLACE

*R*eplaces all occurrences of the first specified parameter with the second specified parameter.

Example: Replaces all commas (first parameter) with a decimal point (second parameter).

Sample mapping function

/EBY/PDBO_MAF_SAMPLE

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.

Insert space before first upper case character

/EBY/PDBO_MAF_SPACE_BUC

Inserts a space before the first upper case character found in the field.

Subtract parameter from value

/EBY/PDBO_MAF_SUBTRACT

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.

Get a system value

/EBY/PDBO_MAF_SYST

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.

Unit of measurement ISO or string to SAP

/EBY/PDBO_MAF_UOM

Converts an ISO unit of measurement or a text string to the SAP internal format required by ReadSoft 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.

Change the value to upper case

/EBY/PDBO_MAF_UPPERCASE

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 ReadSoft PROCESS DIRECTOR applies a preset value. The following standard preset IDs are available.

Preset ID	Applies presets
APPROVE	When users approve a document in a workflow step.
CHECK	Before checks and determinations run.
CREATE	When a new document, a new line item, or another sub-object such as accounting and partner lines, is created.
INSERT	When a newly created document is saved for the first time (that is, inserted in the database).
POST_BEFORE	Before a document is posted.
POST_SUCCESS	When a document is successfully posted.

Preset ID	Applies presets
RECALL	When a workflow is recalled. 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**
/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 Accounts Receivable
- 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.

Duplicate value check

/EBY/CL_PDBO_EVT_DFC_CHK

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.

Check if the requested attachments exist

/EBY/CL_PDBO_EVT_CHK_ATTACHMNT

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.

Check if any item exists

/EBY/CL_PDBO_EVT_CHK_ITEM_EXST

Checks whether line items are present or not, and returns a message if there are no line items. In the initialization parameters, you can specify the logical level that should be checked (for example, Items, Partners, Accounts).

Check template

/EBY/CL_PDBO_EVT_CHK_TEMPL

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.

Check if document is excluded from autoposting

/EBY/CL_PDBO_EVT_APE_CHK

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 if the last note was entered by the current user

/EBY/CL_PDBO_EVT_CHK_LASTNOTE

Checks if the current user added the last note for a document. If the document does not have a note or a different user entered the last note, an error message, which states that a new note is required, is displayed.

Mandatory field check based on field status

/EBY/CL_PDWC_EVT_CHK_FLDST_MND

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:

- Checks 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 Accounts Receivable
- 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 Accounts Receivable
- Determinations for Requisitions

Available process parameters

Configure process parameters

ReadSoft PROCESS DIRECTOR provides an easy way for you to control the behavior of the system for the different processes. No knowledge of ReadSoft 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 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
- Checks
- Posting
- Workflow
- Other

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.
Add field values as attachment comment	Automatically adds an object type-specific field value as an attachment comment to the PROCESS DIRECTOR document. In the parameters, specify the corresponding object type and the field name from which the value should be taken. For more information, see Add field values as attachment comments.
Archiving configuration for Ariba	Specifies the archiving logic for the transfer of FI and MM documents from Ariba to ReadSoft PROCESS DIRECTOR Accounts Payable. You can archive either the XML or PDF files that were created based on the Smart Form templates. For more information, see Configure invoice attachments for Ariba documents.
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.
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 Configure IDoc image attachments.

Process	Description
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. For more information, see Configure Web Application attachment deletion. **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.
Mapping ID for RESCAN process	Enables you to assign a mapping ID for the RESCAN process. For more information, see Configure Rescan.
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.
Sort order of attachments (old -> new)	In the parameters, select the order in which the attachments should be sorted; ascending or descending, by date. For more information, see Sort attachments.
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**.
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**.
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.

Process	Description
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.
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.
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**.
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**.

Process	Description
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 ReadSoft 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.
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.

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.

Available standard documentation objects

The following standard documentation objects are available in ReadSoft PROCESS DIRECTOR for use in messages and workflow emails. Most of these documentation objects include placeholders, which ReadSoft PROCESS DIRECTOR replaces with the relevant values when the message or email that contains the documentation object is sent. You can create and edit documentation objects in transaction `SE61`.

Use objects beginning with */EBY/ICWC* for Accounts Payable documents and objects beginning with */EBY/PDWC* for all other document types.

Messages

Documentation object name	Purpose
/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

Documentation object name	Purpose
/EBY/PDBO_DISPATCH_POST	Inform the creator of a document that the document has been posted.
/EBY/PDMDVM_DISPATCH_POST	Inform the creator of a vendor master data maintenance document that the document has been posted.

Collective emails

Documentation object name	Purpose
/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

Documentation object name	Purpose
/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

Documentation object name	Purpose
/EBY/PDWC_SENT */EBY/ICWC_SENT*	Inform a workflow processor that they have received a document for processing.
/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

Not available for Accounts Payable documents.

Documentation object name	Purpose
/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

Not available for Accounts Payable documents.

Documentation object name	Purpose
/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

Not available for Accounts Payable documents.

Documentation object name	Purpose
/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.
/EBY/PDWC_SHORT_WF_RECL_HTML	Same as */EBY/PDWC_SHORT_WF_RECALLED*, but in HTML format.

Workflow cancelation emails

Not available for Accounts Payable documents.

Documentation object name	Purpose
/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.

Email-based Approval emails

Not available for Accounts Payable documents.

Documentation object name	Purpose
/EBY/PDWC_SENT_MA	Inform an Email-based 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 an Email-based 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 an Email-based Approval workflow processor that their action (approve, reject, add note) was successful.

Documentation object name	Purpose
/EBY/MOBILE_APPR_ERR	Inform the workflow processor that their action (approve, reject, add note) could not be performed.

Other emails

Not available for Accounts Payable documents.

Documentation object name	Purpose
/EBY/PDFI_ARDI_REMINDER	Inform the creator of an FI Deferral (invoice) document that the validity date of the posted document has been exceeded.
/EBY/PDFI_GLPR_REMINDER	Inform the creator of an FI Provision document that the validity date of the posted document has been exceeded.

Available placeholders for documentation objects

You can use placeholders 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.

The following placeholders are available for use in workflow and message texts.

User data

Placeholder	Inserts in the email
&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.

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

Placeholder	Inserts in the email
&HEADER-[FIELD]&	You can use any field provided in the header interface structure of the process 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*, and so on. 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

Placeholder	Inserts in the email
&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.
&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.

Placeholder	Inserts in the email
&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.

Email-based Approval data

Placeholder	Inserts in the email
&MOBILE-APPROVE&	Link that the Email-based 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 Email-based 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.
&MOBILE-NOTE&	Link that the Email-based 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 data

Placeholder	Inserts in the email
&URL&	The Web Application URL specified in Workflow General Settings. 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&*

Placeholder	Inserts in the email
	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**.
&HTML_EMAIL_CSS_STYLES&	Add this alias to the <body> section of your own HTML email documentation objects to use your own style sheet for HTML emails. See Customize HTML emails for more information. **Note**: This placeholder applies only to HTML emails.

Available 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.
/EBY/PDBO_DEBUG_DET	Set to a value of X, this parameter displays detailed messages about the determination search results and result handling. This enables you to debug determinations without having to display all debug messages.
/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

/EBY/PDWC_DUE_DATE_CHECK program

The `/EBY/PDWC_DUE_DATE_CHECK` program 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.

Note: Customers using WORK CYCLE for processing Accounts Payable documents can assign email users to workflow steps. Currently, ReadSoft 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
Document selection	
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.
Collective emails	
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	
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.
Email type	
Collective email	One email containing all workflow step notifications for the processor since the last program run is sent. This means that when this option is selected, all the information related to overdue steps, receipts, as well as approved, rejected, and recalled workflows is sent in that one email.
Email per document	An individual email for each document is sent. When this option is selected, the notification about overdue steps or steps received for processing is sent in single emails, and the other notifications are sent in one collective email per receiver. **Note**: This setting only applies to receipt and reminder emails.
Document batch size	Specifies the number of documents processed in one batch.

Note: In the workflow step configuration, the **Collective email** option in the **Send emails** setting is used only for the generation of receipt emails. For more information about configuring workflow steps, refer to the *PROCESS DIRECTOR Configuration Guide*.

/EBY/PDBO_REPETITOR program

Use the `/EBY/PDBO_REPETITOR` program to automatically check, post, or delete documents at regular intervals. After checking/posting, the document status is automatically changed.

The program can also be called with the `/EBY/PDBO_REPETITOR` transaction.

General selection criteria

Setting	Description
Object type	Process type for which documents will be processed.
PD document number	Document or range of documents to be processed.
Document status	Statuses of the documents to be processed. For example, when checking or posting documents, you should exclude documents that have already been posted.
Substate	Substatuses of the documents to be processed.
Workflow	Workflow statuses of the documents to be processed. 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.
Delete	Select to delete the documents.

Messages and application log

Setting	Description
Document batch size	The number of documents that will be processed at the same time.
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 program

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

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

Download XML file

Setting	Description
To file	Name of the XML file in which the configuration will be stored.
Of document type	The document type to be downloaded. It includes independent tables of all the document types. When a table is document dependent: • 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.

Setting	Description
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.

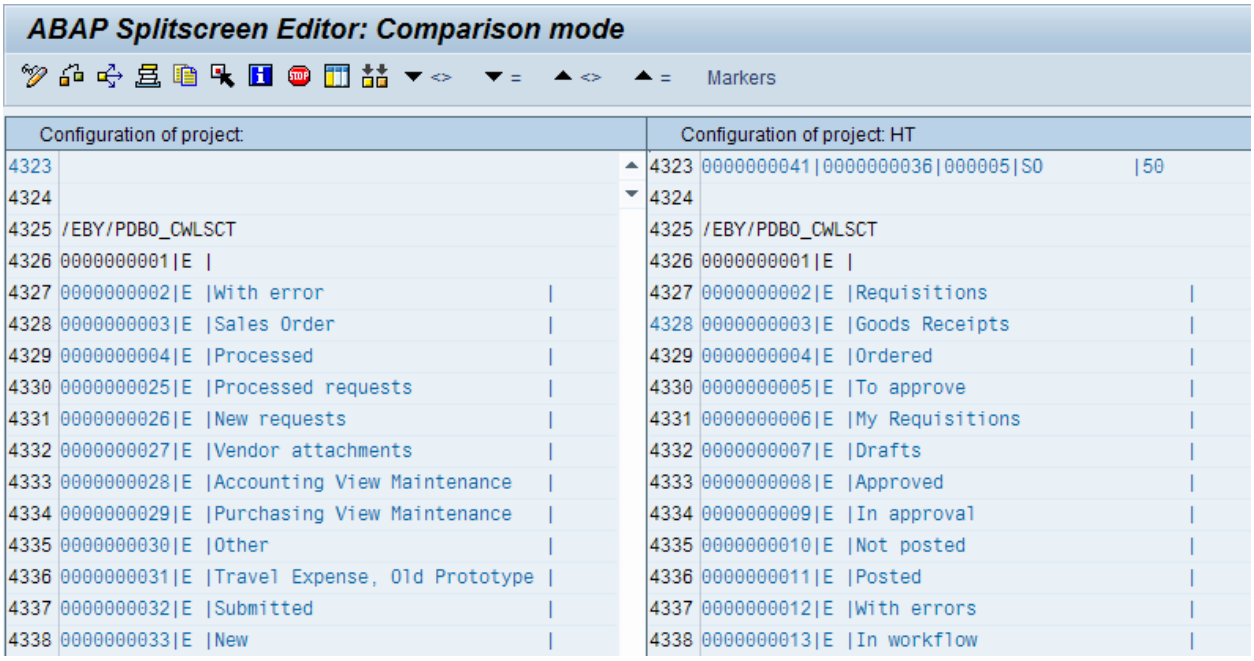
Note: Uploading a Worklist configuration to a project in which the Worklist has already been configured is not recommended.

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.
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.
Ignore wrong table type	Activate this option to allow upload of tables with a delivery class other than C or S, or tables that have delivery class C or S but no CLIENT field. Enter these tables in the **Special tables** field. This feature should be used with care.

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 program

Use the `/EBY/PDBO_CONFIG_GEN_WORKLIST` program to generate the PROCESS DIRECTOR Worklist. You can also use the Worklist configuration screen to manually generate the Worklist.

Parameter	Description
Document batch size	The number of documents processed in one batch.

Remove orphan entries	Removes orphan entries from the table that contains the document assignment.
Calculate document counters	Recalculates Worklist node counters during Worklist regeneration.

/EBY/PDBO_CHECK_NESTEDSET_NEW program

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 program

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 program

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 program

In ReadSoft 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 program

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

/EBY/PDBO_DOC_STATUS_UPDATE program

Use the `/EBY/PDBO_DOC_STATUS_UPDATE` program to check and modify the status, substatus, or workflow status of PROCESS DIRECTOR documents, for any process type. The new status is written directly to the database. You can set any available status value, independent of the allowed status transitions.

Setting	Description
Object type	Process type for which the document status will be checked/modified.
PD document number	The PROCESS DIRECTOR document number.
Current value	The current document status, substatus, or workflow status.
New value	The new value that you want to set for the document status or substatus, or for the workflow status.

/EBY/PDBO_MAF_REGEX_FILTER program

The `/EBY/PDBO_MAF_REGEX_FILTER` program uses a regular expression to filter and extract a substring of an external value and map it to an internal field. The regular expression can be maintained as a mapping function parameter.

When the external string does not match the regular expression, it is deleted. From the regular expression, the first defined submatch specified between () is then used as the external value.

/EBY/PDBO_MASS_FILE_PROCESSING program

The `/EBY/PDBO_MASS_FILE_PROCESSING` program uploads a list of files from a specific directory and posts them in one batch. After posting, the processed files are added to a ZIP file, which is then archived in a directory specified by the user, with the current timestamp as the file name.

Note: The `/EBY/PDBO_MASS_FILE_PROCESSING` program is suitable for background processing.

General selection criteria

Setting	Description
Object type	Process type for which the files will be processed.
File upload ID	The file upload ID.

Source folder

Setting	Description
Source directory	The path on the local machine or application server from which the files need to be loaded.

Target folder

Setting	Description
Target directory	The directory in which the ZIP archive will be stored for further processing.

Folder Type

Setting	Description
Local drive	Select this option if the folder that needs to be processed resides on your local drive.
Application server	Select this option if the folder that needs to be processed resides on the application server.

Options for source files

Setting	Description
Delete processed files	Select to remove the processed files from the source directory after processing.
Test run	Select to simulate the uploading and posting process, without actually processing the files.
Create application log	Select to create an application log that you can view in the SLG1 transaction.
Remove directory lock	To prevent parallel processing of files, on process startup, the source directory is locked. If, due to an error, the lock is not removed at the end of the processing run, select this check box to unlock the directory for the next run.

/EBY/PDBO_RFC_DATA_RECEIVE program

Use the `/EBY/PDBO_RFC_DATA_RECEIVE` program to submit invoice data. The submitted PROCESS DIRECTOR Accounts Payable documents will then be accessible using the `/COCKPIT/1` transaction.

- With this program, you can submit the invoices by using the object type, *IV*.
- The transferred data is directly stored in the `/COCKPIT/*` tables, or through the `/COCKPIT/RFC_DATA_SUBMIT` program.
- The program returns the GUID, status, and messages.

You can also use the `/EBY/PDBO_RFC_DATA_RECEIVE` program to submit PROCESS DIRECTOR Accounts Payable OCR information for One Touch Processing (OTP).

- You can submit the OCR data by using a new object type (for example, *OC* for OCR data).
- The transferred data is submitted to `/COCKPIT/RFC_OTP_OCR_DATA_SET`.
- Information from `/COCKPIT/RFC_OTP_OCR_DATA_SET` is returned in the `/EBY/PDBO_RFC_DATA_RECEIVE` program.

See the *PROCESS DIRECTOR Accounts Payable Configuration Guide* for more information.

/EBY/WC_DELETE program

The `/EBY/WC_DELETE` program deletes workflow data for a specific document. You can use this program to remove a document from workflow, for example, if users cannot recall the document because the workflow is no longer valid due to changes in the workflow customizing.

You can also run the program with transaction `/EBY/WC_DEL`.

Setting	Description
PD document number	The PROCESS DIRECTOR document number.
Process type	The PROCESS DIRECTOR process type.
PD document status	Select this option to view the document's workflow status.
Delete workflow	Select this option to delete the document's workflow data.

/EBY/WC_SUBST_MANAGER program

Use the `/EBY/WC_SUBST_MANAGER` program to globally manage substitutes for SAP and WORK CYCLE users. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

User Exits/BAdIs

ReadSoft 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.

Basic

BAdI	Description
Screen PBO/PAI	Add custom logic for the screen display. For example, you can use this BAdI to display or verify values.
Modify Worklist	Add custom logic for the PROCESS DIRECTOR Worklist configuration.
Modify field profiles	Add custom logic when field statuses are applied.

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

Web Application

BAdI	Description
Set messages	Add custom logic to modify messages sent to the Web Application.
Set actions	Add custom logic to modify which 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.

Workflow

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

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

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

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

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.

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** 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

Change 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.

Change 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
	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
	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.

ReadSoft PROCESS DIRECTOR provides a standard implementation for Accounts Payable, */EBY/ICWC_USER_LOGON*, which should be activated if ReadSoft 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

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
	VALUE(IC_PASSWORD)	PROCESS DIRECTOR user password type
	VALUE(IC_NEW_PASSWORD)	PROCESS DIRECTOR user password type for new password
	CS_MESSAGE	Return parameter

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
	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

Type	Parameter	Description
	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

Type	Parameter	Description
	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 ReadSoft 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 ReadSoft 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

Type	Parameter	Remarks
	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
	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.
	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

Type	Parameter	Remarks
	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 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

Type	Parameter	Description
	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.
	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.
	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 or rejected. 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
	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

Type	Parameter	Remarks
	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.

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
	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	Description
	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
	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.
	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

Type	Parameter	Remarks
	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.
	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

Accounts Payable (IV)

Presets for Accounts Payable

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

Try to post the IV document

/EBY/CL_ICIV_EVT_PRESET_POST

Checks for a corresponding Accounts Payable document and its status. If that document is in workflow, an attempt to approve it as the current user is triggered. If it is not in workflow and has not yet been posted, the posting action is triggered.

This preset can be used for any PROCESS DIRECTOR document type.

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 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 Configuration Guide* for more information.

GUI area	Comp. type	Grid No.	Logical level
SAP GUI header data	SAP GUI	1	HEADER
SAP GUI line items grid	SAP GUI	2	ITEMS
SAP GUI Accounts grid	SAP GUI	3	ACCOUNT
SAP GUI PO Items grid	SAP GUI	4	POITEM
	A1 Additional grid 01	1	TAX
	A1 Additional grid 01	2	ACCASS
	A1 Additional grid 01	4	PO_ACCASS

GUI area	Comp. type	Grid No.	Logical level
	WA Document Detail	1	HEADER
	WA Document Detail	2	ITEM
	WA Document Detail	3	TAX
	WA Document Detail	4	ACCOUNT
	WA Document Detail	5	ACCASS
	WA Document Detail	6	SORDER
	WA Document Detail	7	PO_ACCASS
	WA List Header	1	HEADER
	Workflow status	1	HEADER
	Header data for archive log	1	HEADER

Programs

/EBY/ICIV_DOC_SYNCHRONIZE program

The `/EBY/ICIV_DOC_SYNCHRONIZE` program registers existing PROCESS DIRECTOR Accounts Payable documents in ReadSoft PROCESS DIRECTOR. You must run this program so that users can view and access PROCESS DIRECTOR Accounts Payable documents in ReadSoft PROCESS DIRECTOR.

You can specify which documents to register. For example, you may want to exclude documents that are already complete.

Note: This program does not support Fast Entry documents.

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`).
Document date	Document date of the document (*DOC_DATE*).
Creation date	Date on which the document was transferred to ReadSoft PROCESS DIRECTOR (*CR_DATE*). **Note**: Time zones (system or user setting) in the remote systems must match the umbrella system, otherwise selection by creation date will not return the correct results.

Setting	Description
Company code	Company code for which documents should be registered.
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 program

The `/EBY/ICIV_ACCRUAL_REPORT` program collects 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. Users 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. If the Financial Postings process type is not installed, users can process items manually. Users can also create the report with transaction `/EBY/ICIV_ACCR_REP`, provided they have authorization for this transaction.

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

/EBY/ICIV_VAT_REPORT program

The `/EBY/ICIV_VAT_REPORT` program collects tax information from all un-posted PROCESS DIRECTOR Accounts Payable documents and presents it in a VAT report. The report summarizes the information by currency and tax code, and by gross and net amount per currency. You can also create a VAT report using the `/EBY/ICIV_VAT_REP` transaction.

/EBY/ICIV_DOC_MAPPING_REGEN

Use the `/EBY/ICIV_DOC_MAPPING_REGEN` program to regenerate the mapping of GUIDs from ReadSoft PROCESS DIRECTOR Accounts Payable to ReadSoft PROCESS DIRECTOR.

Setting	Description
Identification	Unique document identifier (GUID).
Document number	PROCESS DIRECTOR document number.
System ID	System for which regeneration should be performed.

Accounts Receivable (PA)

Presets for Accounts Receivable

The following standard presets are available for Accounts Receivable. 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 discount and gross amount from items

/EBY/CL_PDPA_PRESET_DISCOUNT

Calculates the **Gross amount** and **Discount** at header level from the line item data. The **Gross amount** is the sum of the **Payment amount** plus discounts at item level plus deductions at item level. The preset only applies if the **Discount** field is initial and the **Gross amount** field is initial or equal to the **Payment amount**.

Preset item data from BSID

/EBY/CL_PDPA_PRESET_ITEM_BSID

Presets the following item data from the *BSID* table.

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

The preset also populates the customer number in the header data 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, taking the discount into account.

- **Payment amount** = **Gross amount** minus **Cash discount amount**
- **Gross amount** = **Payment amount** plus **Cash discount 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, you must add it to the Accounts Receivable presets configuration, not the Electronic Bank Statement's presets configuration.

Checks for Accounts Receivable

The following standard checks are available for Accounts Receivable. 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 applies, the check displays a message.

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. The **Payment amount** must be equal to **Gross amount** minus **Discount amount** minus **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.

Checks if the document is a duplicate of a previous one

/EBY/CL_PDPA_EVT_CHK_DUPLICATE

Checks if a payment advice with the same combination of company code, customer, payment amount, and payment date entered by the user is already available in the database. If yes, an error message about a potential duplicate document existing, is displayed.

Note: Rejected documents are not taken into account.

Check open item is entered multiple times

/EBY/CL_PDPA_EVT_CHK_ITM_MULTI

Checks if an open item has been entered multiple times for the same document. If yes, an error message with the corresponding document number is displayed for each duplicate line item.

Determinations for Accounts Receivable

The following standard determinations are available for Accounts Receivable. Note that if allowed payers are defined in the customer master, the determinations also take these allowed payers into account.

Payment Advice customer determination

/EBY/CL_PDPA_DET_CUST_INIT

Determines the customer number based on the following data, 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*

Search type	Search name
Reference document number	*/EBY/CL_PDPA_DET_CUST_XBLNR*
Reference document number from reference in item	*/EBY/CL_PDPA_DET_CUST_XR_XBLNR*

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 the following data, 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*
Sales order. By default, the determination performs the search on the **Reference**, **DocumentNo** and **Billing Doc.** fields. You can specify other fields in the **Mapping of BSID, BSAD and BSEG fields to PDPA items** search parameters.	*/EBY/CL_PDPA_DET_ITEMMAP_VBELV*

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 to retrieve.

Depending on the defined searches, the determination adds the following items.

Propose all items

/EBY/CL_PDPA_DET_ITEMPROP or */EBY/CL_PDPA_DET_ITEMPROP_Z00*

Proposes all open items.

From invoice list - having the same reference

/EBY/CL_PDPA_DET_ITEMPROP_Z02

Proposes only items from the SAP Invoice List that have the same reference number from the **Payment document number** field.

Currently due items

/EBY/CL_PDPA_DET_ITEMPROP_Z21

Currently due items are items for which the invoice due date is later than the payment date, and the posting date is before the payment date by a specified minimum number of days, which you define in the search parameters **Min time** field.

Currently due items - payment amount preferred

IEBY/CL_PDPA_DET_ITEMPROP_Z211

Same as **Currently due items**, but in addition to dates, the search also takes the payment amount into account. If there are currently due items that match the payment amount, the determination proposes only these items. If there are no currently due items that match the payment amount, the determination proposes all currently due items.

Overdue items

/EBY/CL_PDPA_DET_ITEMPROP_Z22

Items for which payment is overdue.

Overdue items - payment amount preferred

/EBY/CL_PDPA_DET_ITEMPROP_Z221

Same as **Overdue items**, but in addition to dates, the search also takes the payment amount into account. If there are overdue items that match the payment amount, the determination proposes only these items. If there are no overdue items that match the payment amount, the determination proposes all overdue items.

Unique combination of items

IEBY/CL_PDPA_DET_ITEMPROP_Z231

The determination only adds items if there is a single, unique combination of open items that matches the payment amount.

The following example illustrates a unique combination.

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.

Unique combination with single match first

/EBY/CL_PDPA_DET_ITEMPROP_Z231

Same as **Unique combination**, except that as soon as a combination is found that matches the payment amount, the search does not take other open items and the determination proposes the first match found.

Handle discrepancy

/EBY/CL_PDPA_DET_ITEMPROP_Z24

*D*o not add this search to the configuration. PROCESS DIRECTOR uses this search in the background for the discrepancy handling functionality.

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 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*
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 Account Receivable. See the *PROCESS DIRECTOR SAP Configuration Guide* for more information.

GUI area	Comp. type	Grid No.	Logical level
SAP GUI document list	SAP GUI	1	HEADER
SAP GUI line items grid	SAP GUI	2	ITEMS
Web Application document detail header data	WA Document Detail	1	HEADER
Web Application document detail line items grid	WA Document Detail	2	ITEMS
Web Application document list	WA List Header	1	HEADER
Workflow status dialog box	Workflow status	1	HEADER
Workflow log document	Header data for archive log	1	HEADER

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 ...
	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

/EBY/PDPA_FINALIZE_DOC program

The `/EBY/PDPA_FINALIZE_DOC` program links 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. The program ignores items that are not assigned.

Asset Acquisitions (AA)

Checks for Asset Acquisitions

The following standard check is available for Asset Acquisitions. 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 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 Configuration Guide* for more information.

GUI area	Comp. type	Grid No.	Logical level
SAP document list	SAP GUI	1	HEADER
SAP List of Assets grid	SAP GUI	2	List of Assets
Web Application document detail header data	WA Document Detail	1	HEADER
Web Application document detail List of Assets grid	WA Document Detail	2	List of Assets
Web Application document list	WA List Header	1	HEADER
Workflow status popup	Workflow status	1	HEADER
Archive log	Header data for archive log	1	HEADER

Asset Retirement (AR)

Presets for Asset Retirement

The following standard preset is available for Asset Retirement. You can also use general presets 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 standard check is available for Asset Retirement. 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 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 Configuration Guide* for more information.

GUI area	Comp. type	Grid No.	Logical level
SAP document list	SAP GUI	1	HEADER
SAP List of Assets grid	SAP GUI	2	List of Assets
Web Application document detail header data	WA Document Detail	1	HEADER
Web Application document detail List of Assets grid	WA Document Detail	2	List of Assets
Web Application document list	WA List Header	1	HEADER
Workflow status popup	Workflow status	1	HEADER
Archive log	Header data for archive log	1	HEADER

Asset Transfer Posting (AT)

Presets for Asset Transfer

The following standard preset is available for Asset Transfer. 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 standard check is available for Asset Transfer. 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 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 Configuration Guide* for more information.

GUI area	Comp. type	Grid No.	Logical level
SAP document list	SAP GUI	1	HEADER
SAP List of Assets grid	SAP GUI	2	List of Assets
Web Application document detail header data	WA Document Detail	1	HEADER
Web Application document detail **List of Assets** grid	WA Document Detail	2	List of Assets
Web Application document list	WA List Header	1	HEADER
Workflow status popup	Workflow status	1	HEADER
Archive log	Header data for archive log	1	HEADER

Customer Orders (SO)

Checks for Customer Orders

The following checks are available for Customer Orders. You can also use general checks that are available for all process types.

Each check has one of the following check types assigned.

- P (simulate)
- S (standard)
- R (result)

If several checks are configured for the same action, a simulation check has to be performed prior to a result check. This is achieved by assigning a lower suborder number to the simulation check, than what is assigned to the result check. If no simulation check is performed, when the check configuration is saved, a warning message is displayed.

Check count of order item positions

/EBY/CL_PDSO_EVT_CHK_ITEMS

Check type: S (standard)

Checks for a required number of order line items or a maximum permitted number of line items. Also checks whether a material number is 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 does not apply, or if the material number is missing in any of the line items, the check displays a message.

Check if order is in reference to another document

/EBY/CL_PDSO_EVT_CHK_REFDOC

Check type: S (standard)

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

If the reference could be a contract, configure the contract relation determination to verify the contract reference.

Check if sold-to party has open documents

/EBY/CL_PDSO_EVT_CHK_OPEN

Check type: S (standard)

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 using the **PROCESS DIRECTOR SO open document check** option (*/EBY/PDSO_LEVTP_CHK_OPEN*).

You can use this check to prevent automatic posting in systems where it is common to work with quotations, but the quotation number for the sales order is not captured.

Check incompletion log

/EBY/CL_PDSO_EVT_CHK_INCOMP

Check type: R (result)

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

Check type: R (result)

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

Check type: R (result)

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 using the **Parameters check working days deviance** option (*/EBY/PDBO_SEVTP_WORKING_DAYS*). 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

Check type: S (standard)

Checks mandatory fields for special order types. You can specify the document types to check in the initialization parameters using the **PROCESS DIRECTOR SO ordertype mandatory field checks** option (*/EBY/PDSO_LEVTP_CHK_MANDFLDS*).

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

Check type: S (standard)

Checks the Listing/Exclusion rules configured in SAP, which enable you to 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 the *KOMKG* structure includes additional header fields for Listing/Exclusion, such as customer-specific fields, and these fields should also be used for the check, you must map these fields to the corresponding PROCESS DIRECTOR fields. You can define this field mapping in the check's initialization parameters using the Map header field to KOMKG field option (*/EBY/PDSO_LEVTP_HDR2KOMKG_MAP*).

Check minimum order quantity for all items

/EBY/CL_PDSO_EVT_CHK_MIN_ORD_Q

Check type: S (standard)

Checks whether the minimum order quantity is reached for every line item. The checks takes different order units into account.

Check price discrepancies on item level

/EBY/CL_PDSO_EVT_CHK_ITEMPRICE

Check type: R (result)

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 using the **Parameters for currency value difference limits** option (*/EBY/PDBO_SEVTP_CHK_CURRVAL*).

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

If one of the defined rules applies, the check displays a message.

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

Check type: R (result)

Checks for sales-related texts, such as following types of text.

- Customer information
- Material information
- Order type related information

In the initialization parameters you can define how the texts are displayed using the **Parameters for check of texts** option (*/EBY/PDSO_SEVTP_CHK_TEXTS*).

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.
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

Check type: P (simulate)

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.

In the initialization parameters, you can specify options for the simulation using the **Parameters for event create_from_s** option (*/EBY/PDSO_SEVTP_CREATE*).

/EBY/CL_PDSO_EVT_CHK_KUN_DEL

Check type: S (standard)

Checks whether the given business partners are marked for deletion in SAP.

Check whether PO number has been already used

/EBY/CL_PDSO_EVT_CHK_DUPLICATE

Check type: S (standard)

Checks whether the number in the **PO number** field has already been used. The check first checks PROCESS DIRECTOR documents, then all posted SAP orders.

In the initialization parameters, you can specify options for the check using the **Parameters for even check duplicate** option (*/EBY/PDSO_SEVTP_CHK_DUPLICATE*).

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.
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, post the current document as a new customer order in SAP, manually update the customer order in SAP, or delete the PROCESS DIRECTOR document. See the *PROCESS DIRECTOR User Guide* for more information.

Check scheduled quantity against item quantity

/EBY/CL_PDSO_EVT_CHK_SCHDL_QTY

Check type: S (standard)

Checks whether the sum of the scheduled quantities for a line item match the ordered quantity.

Compare header net values of document with simulated order

/EBY/CL_PDSO_EVT_CHK_HDRNETVAL

Check type: R (result)

Check an expected order value against defined ranges.

In the initialization parameters, you can define the following rules using the **Parameters for currency value difference limits** option (*/EBY/PDBO_SEVTP_CHK_CURRVAL*).

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

If one of the defined rules applies, the check displays a message.

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

Check type: R (result)

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

The following standard determinations are available for Customer Orders.

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*
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 You can 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*
Name from the list of all maintained contacts of a given business partner (Sold-To, Ship-To, Bill-To). You can specify the partner function in the search parameters.	*/EBY/CL_PDSO_DET_CONTACT_P_N*

Search area	Search name
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*
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*

Search type	Search name
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 using the Field to be used for material determination option (*/EBY/PDSO_SEVTP_MAT_FIELD*).

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.

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 the following master data, 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*

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 You can 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)
Business partners You can 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

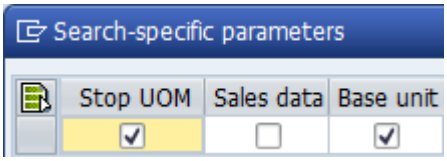
/EBY/CL_PDSO_DET_UOM_INIT

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 the following parameters.

- **PROCESS DIRECTOR SO partner determination parameters**
- Replaces characters while searching.
- **PROCESS DIRECTOR SO partner excluded number range**
- Excludes a specific number range from the search.
- **Validate entry against KNVP table**
- Checks the *KNVP* table for an existing sold-to number for the respective sales organization level.

Process parameters for Customer Orders

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

Process	Description
Posting parameters (Posting)	Enables you to override the default settings for the posting of customer orders, by defining customer parameters for the following item lists. • **Parameters for event chk_create - Conditions** Options for simulating the creation of customer orders in SAP • **Parameters for event chk_create - Auto. sched.** Options for the automatic generation of missing schedule lines • **Parameters for event create_from_s** Options for simulating the posting of customer orders

Parameters for event chk_create - Conditions item list

Parameter	Description
Sales Document Type	The type of the sales document. If this field is empty, the condition is applied to all the sales document types.
Condition type	The key that uniquely identifies the condition type that should be applied.
Use net value	Select the this check box to use the net value of the line items for the condition. If you do not select it, the price value is used.
Field Name	The name of the field that should be used as the value for the condition type. **Note**: Any field from the `/EBY/PDSO_TITM` table, which has the decimal data type assigned, can be used.

Parameters for event chk_create - Auto. sched. item list

Parameter	Description
Sales Document Type	The type of the sales document. If this field is empty, the condition is applied to all the sales document types.
No auto. schedule line generation	Disables the automatic generation of missing schedule lines.

Parameters for event create_from_s item list

Parameter	Description
Binary rel. type	The binary relationship type that is needed when writing object references.
Internal Item Number Assignment	Activates the internal assignment of the item numbers.
Error Handling	Allows you to create a customer order, even if some of the items cannot be created.
Conversion of Partner Function and Order Type	Checks whether the customizing has been done in such a way that the abbreviations of the order type and partner function can be converted from another language into German.

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 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*

Logical level	Package	Interface structure	Customer Includes
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. ReadSoft 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*
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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
SAP GUI	2	ITEMS
SAP GUI	3	HEADER_PARTNERS
Additional grid 01	1	HEADER_COND
Additional grid 01	2	ITEM_COND
Additional grid 01	3	SCHEDULES
WA Document Detail	1	HEADER
WA Document Detail	2	ITEMS
WA Document Detail	3	HEADER_PARTNERS

Comp. type	Grid No.	Logical level
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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	SO items
	IT_HDR_CONDITIONS	SO header conditions
	IT_HDR_PARTNERS	SO header partners
	IT_ITEM_CONDITIONS	SO item conditions
	IT_ITEM_PARTNERS	SO item partners
	IT_ITEM_SCHEDULES	SO item schedules
	IS_EVENT	Configuration of processes
	CT_CONTRACTS	Table of BAPICTR

Type	Parameter	Description
	CS_HEADER	SO BAPI header
	CT_ITEMS	SO BAPI items
	CT_PARTNERS	SO BAPI partners
	CT_CONDITIONS	SO BAPI conditions
	CT_SCHEDULES	SO BAPI schedules
	CT_TEXTS	SO BAPI texts
	CT_EXTENSIONS	SO 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
	CT_CONF_VALUE	Table of BAPICUVAL
	CT_CONF_PARTOF	Table of BAPICUPRT
	CT_CONF_INST	Table of BAPICUINS

Type	Parameter	Description
	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
	CC_OBJ	Object type
	CC_WHEN_ERROR	Single-character flag
	CC_CALL_ACTIVE	Not More Closely Defined Area, Possibly Used for Patchlevels
	CT_CMPB_ASGN	Table of BAPIBTSEL
	CS_SWITCH	SD Checkbox for the Logic Switch. This corresponds to the LOGIC_SWITCH parameter of the function module SD_SALESDOCUMENT_CREATE and allows you to change these fields according to customer needs.

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	Sales order
	IS_HEADER	SO BAPI header
	IT_ITEMS	SO BAPI items
	IT_PARTNERS	SO BAPI partners

Type	Parameter	Description
	IT_CONDITIONS	SO BAPI conditions
	IT_SCHEDULES	SO BAPI schedules
	IT_TEXTS	SO BAPI texts
	IT_EXTENSIONS	SO BAPI Extension
	IT_BUSINESS	Table of BAPISDBUSI
	IT_KONV	Table type for konv
	CS_BO_HDR	Data fields of a PD base object
	CS_SO_HDR	Persistent sales order header data
	CT_HDR_CONDITIONS	SO header conditions
	CT_HDR_PARTNERS	SO header partners
	CT_ITEMS	SO items
	CT_ITEM_CONDITIONS	SO item conditions
	CT_ITEM_PARTNERS	SO item partners
	CT_ITEM_SCHEDULES	SO item schedules

Electronic Bank Statements (ES)

Presets for Electronic Bank Statements

The following standard presets are available for Electronic Bank Statements. 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, you must add it to the Accounts Receivable presets configuration, not the Electronic Bank Statement presets configuration.

Update items in ES by checking area 2

/EBY/CL_PDES_PRESET_EPERL

Sets the status of an Electronic Bank Statement line item to **Cleared** if the **Posting Area 2** information in FEBAN has been filled, but not **Posting Area 1**.

Update items in ES by clearing info

/EBY/CL_PDES_PRESET_FEBEP

Checks if clearing has been performed outside of the FEBAN transaction. If clearing postings are found, the **Posting Area 1** data and the **Posting Area 2** data is updated accordingly in the *FEBEP* and PROCESS DIRECTOR tables.

Determinations for Electronic Bank Statements

The following standard determination is available for Electronic Bank Statements.

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*
Search by item key	*/EBY/CL_PDES_DET_PDPA_ITM_KEY*
Search by value	*/EBY/CL_PDES_DET_PDPA_VALUE*

In the search 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 applies, the determination displays a message.

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 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. ReadSoft 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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
SAP GUI	2	ITEMS
SAP GUI	4	PA_ITEMS
WA Document Detail	1	HEADER
WA Document Detail	2	ITEMS
WA Document Detail	4	PA_ITEMS
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

Financial Postings (FI)

Presets for Financial Postings

The following standard presets are available for Financial Postings. 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 in new positions.

Note: This preset is only for Customer Postings.

Preset for payment terms

/EBY/CL_PDFI_PRESETS_PAY_TERMS

Automatically populates the payment terms fields from the specified customer and company code.

Note: This preset is only for Customer Postings.

Checks for Financial Postings

The following standard checks are available for Financial Postings. 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.

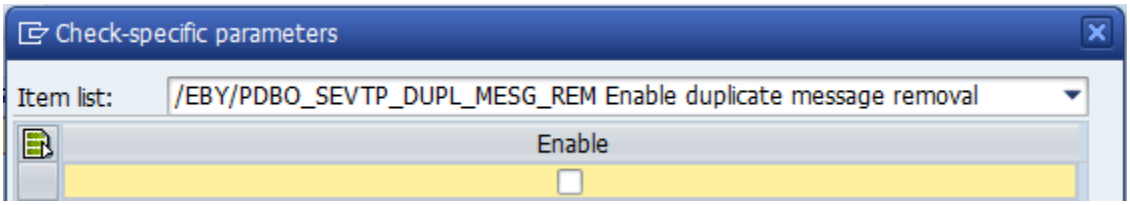
This check is pre-configured in the FI document creation process and does not have to be added to the checks configuration.

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.

In the initialization parameters, you can enable the removal of duplicate messages.



You can also specify exceptions, that is, allow certain messages to be displayed more than once.

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
Posting parameters (Posting)	Enables you to suppress duplicate messages returned by the BAPI. In the parameters, select **Enable duplicate message removal** and check the **Enable** check box. To specify exceptions (that is, to allow certain messages to be displayed more than once), in the parameters, select **Restrictions for message duplicate removal** and add the message type, message ID and message number.

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 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. ReadSoft 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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
SAP GUI	2	ITEMS
WA Document Detail	1	HEADER
WA Document Detail	2	ITEMS

Comp. type	Grid No.	Logical level
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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

Type	Parameter	Description
	CT_ACCOUNTTAX	Tax item
	CT_CURRENCYAMOUNT	Currency items
	CT_CRITERIA	CO-PA account assignment characteristics
	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
	CB_CHECK	Only check
	CC_UPDATEMODE	Update mode for call transaction

Type	Parameter	Description
	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
	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

Type	Parameter	Description
	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
	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

/EBY/PDFI_GEN_RECURRING_DOCS program

The `/EBY/PDFI_GEN_RECURRING_DOCS` program generates PROCESS DIRECTOR Financial Posting documents from recurring entries. See the *PROCESS DIRECTOR User Guide* or *PROCESS DIRECTOR Web Application User Guide* for information on recurring entries.

Setting	Description
Company code	Company code for which the program creates documents.
Posting type	PROCESS DIRECTOR posting type for which the program creates documents.
With approved workflow only	Generates only documents for which workflow approval is complete.
Display messages	Displays messages generated by the program.
Create application log	Creates an application log. To view the application log, in `SLG1`, select the object **/EBY/PD** and the subjobject **/EBY/PD_FI_RECURRING**.

/EBY/PDFI_REMINDER program

The `/EBY/PDFI_REMINDER` program sends reminder emails to users who create a **Provision** or a **Deferral (invoice)** document when the document exceeds its validity date. ReadSoft PROCESS DIRECTOR provides standard documentation objects to use as texts for reminder emails.

Setting	Description
Company code	Company code for which the program sends emails.
Posting type	Posting type for which the program sends emails.
Sender	Name of the user that appears in the email as the sender.
Sender user type	User type of the user that appears in the email as the sender.
Display messages	Displays messages generated by the program.
Create application log	Creates an application log. To view the application log, in `SLG1`, select the object **/EBY/PD** and the subjobject **/EBY/PD_FI_REMINDER**.

Generic archiving (ARCH_BUS)

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 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.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
WA Document Detail	1	HEADER
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

Goods Receipts (DN)

Presets 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.

Preset for movement type

/EBY/CL_PDDN_PRESET_MOVE_TYPE

Presets the movement type to *101* for goods receipts with a reference (delivery note or inbound delivery) and to *501* for goods receipts without a reference. If other movement types should be used, make a copy of this preset and adjust the coding accordingly.

Checks for Goods Receipts

The following standard checks are available for Goods Receipts. 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

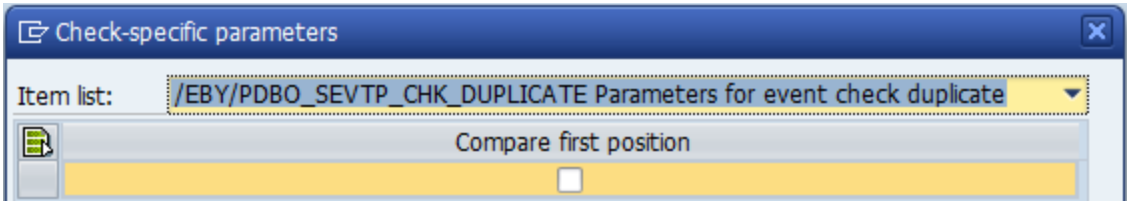
Simulates the creation of the goods movement in SAP and returns all SAP messages generated during the simulation.

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

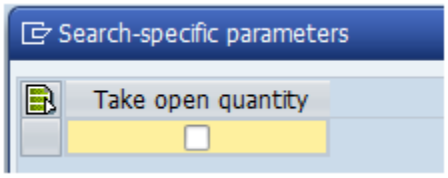
The following standard determinations are available for Goods Receipts.

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*

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*
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*

Search area	Search name
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 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*

Logical level	Package	Interface structure	Customer Includes
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. ReadSoft 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*
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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
SAP GUI	2	ITEMS
SAP GUI	3	PO_ITEMS
Additional grid 01	1	SERIALS
WA Document Detail	1	HEADER
WA Document Detail	2	ITEMS
WA Document Detail	3	PO_ITEMS
WA List Header	1	HEADER

Comp. type	Grid No.	Logical level
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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
	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

Type	Parameter	Description
	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
	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

BAdI implementation /EBY/PDDN_MB_BAPI_GOODSMVT_CRE

By default, when a Goods Receipt is created in the Web Application and posted to SAP, the Web Application RFC user is entered as the creator of the material document (**Entered by** field on the **Doc. info** tab). ReadSoft PROCESS DIRECTOR provides the BAdI implementation */EBY/PDDN_MB_BAPI_GOODSMVT_CRE* to enter the name of the actual user that created the Goods Receipt instead of the RFC user.

The BAdI implementation (or a copy of it) must be activated in SE19 and the runtime behavior set to **Implementation is active**.

Note: SAP Note 1532255 - **BAdI MB_BAPI_GOODSMVT_CREATE can only be implemented once** is required.

Invoice Block and Cancelation (IB)

Checks for Invoice Block/Cancelation

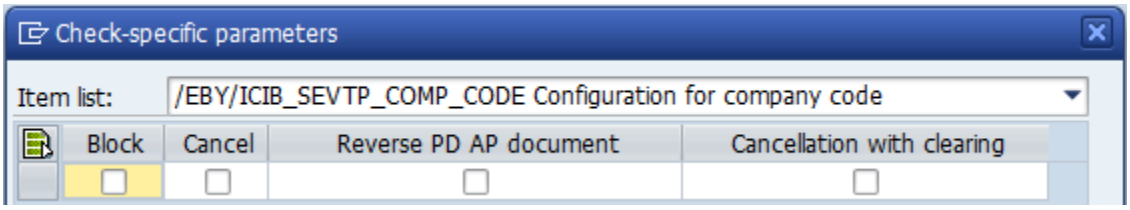
The following standard checks are available for Invoice Block/Cancelation. You can also use general checks that are available for all process types.

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.



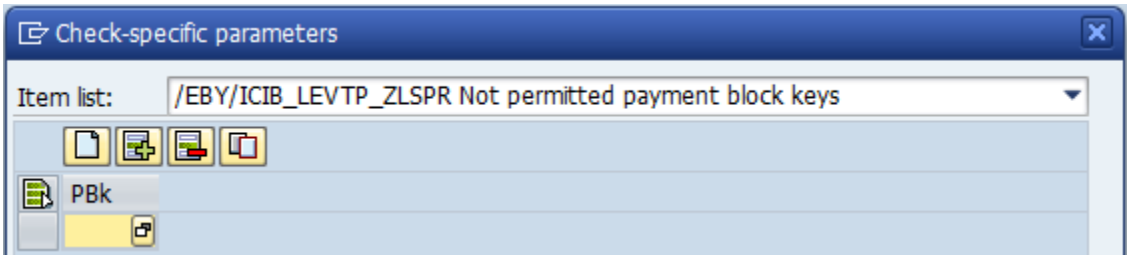
Initialization parameter	Description
Block	Allows invoice blocking.
Cancel	Allows invoice cancelation.
Reverse PD AP document	Allows reversal of the PROCESS DIRECTOR Accounts Payable document.
Cancelation with clearing	Allows cancelation with clearing.

Check for a valid payment block key

/EBY/CL/_ICIB_EVT_CHK_ZLSPR

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 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 Configuration Guide* for more information.

GUI area	Comp. type	Grid No.	Logical level
SAP document list	SAP GUI	1	HEADER
Web Application document detail header data	WA Document Detail	1	HEADER
Web Application document list	WA List Header	1	HEADER

GUI area	Comp. type	Grid No.	Logical level
Workflow status popup	Workflow status	1	HEADER
Archive log	Header data for archive log	1	HEADER

Master Data Maintenance (MD)

Presets for Master Data Maintenance

The following standard presets are available for Master Data Maintenance. 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 standard checks are available for Master Data Maintenance. 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 or 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 or 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 checks whether the email address is valid.

Check IBAN validity

/EBY/CL_PDMDCM_EVT_CHK_IBAN

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

Check VAT number

/EBY/CL_PDMDCM_EVT_CHK_VAT

Checks whether the given VAT registration number is valid. The SAP VAT registration number check must be active for the country (SAP Customizing IMG > General settings > Set Countries > Set Country-Specific Checks).

Check whether an SAP customer master can be created by API

/EBY/CL_PDMDCM_EVT_CHK_API_CRE

Checks whether the customer master data can be posted using the standard SAP posting procedure or not, and returns a message if errors or warnings occur.

Note: If you activate this check and the main memory buffer for the DEBITOR number range object is also active, the number ranges will contain gaps. To generate customer numbers without gaps, you can either use the SNRO transaction to deactivate the main memory buffer, or you can activate parallel buffering. For more information, see SAP note 639754 in the *PROCESS DIRECTOR Import Guide*.

Check withholding tax

/EBY/CL_PDMDCM_EVT_CHK_WHT

Checks the withholding tax.

Sales area existing 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 or 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 or 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 checks whether the email address is valid.

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 VAT number

/EBY/CL_PDMDVM_EVT_CHK_VAT

Checks whether the given VAT registration number is valid.

Note: The SAP VAT registration number check must be active for the country (SAP Customizing IMG > General settings > Set Countries > Set Country-Specific Checks).

Check whether an SAP vendor master can be created by API

/EBY/CL_PDMDVM_EVT_CHK_API_CRE

Checks whether the vendor master data can be posted using the standard SAP posting procedure or not, and returns a message if errors or warnings occur.

The following SAP notes are required: 1733326, 1877573, and 1669240. For more information, see the *PROCESS DIRECTOR Import Guide*.

Note: If you activate this check and the main memory buffer for the KREDITOR number range object is also active, the number ranges will contain gaps. To generate vendor numbers without gaps, you can either use the SNRO transaction to deactivate the main memory buffer, or you can activate parallel buffering. For more information, see SAP note 639754 in the *PROCESS DIRECTOR Import Guide*.

Check withholding tax

/EBY/CL_PDMDVM_EVT_CHK_WHT

Checks the withholding tax.

Check whether an SAP vendor master can be created by API

/EBY/CL_PDMDVM_EVT_CHK_API_CRE

Checks

Process parameters for Vendor Master Maintenance

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

Process	Description
Posting parameters (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 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*
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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
WA Document Detail	1	HEADER
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
WA Document Detail	1	HEADER
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
SAP GUI	2	KEYWORD
SAP GUI	3	TRANSLATION
WA Document Detail	1	HEADER
WA Document Detail	2	KEYWORD
WA Document Detail	3	TRANSLATION
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
WA Document Detail	1	HEADER
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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 Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
SAP GUI	2	BANK
WA Document Detail	1	HEADER
WA Document Detail	2	BANK
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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

Type	Parameter	Description
	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)

Checks for Order Confirmations

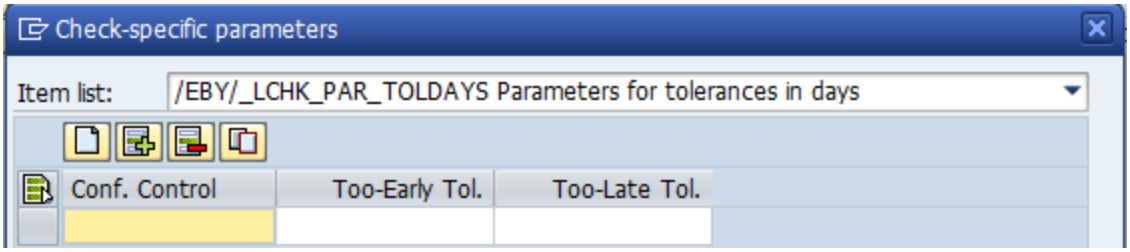
The following standard checks are available for Order Confirmations. You can also use the general checks that are available for all the process types.

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.



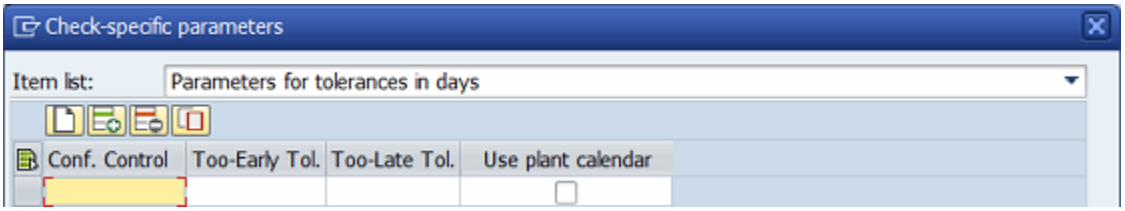
Check confirmed delivery date against previous confirmation

/EBY/CL_PDOR_EVT_CHK_PREV_DATE

Checks the delivery dates of the order confirmation line items against the previously confirmed delivery dates of the corresponding purchase order line items.

This check should have a lower **Suborder** than the **Check confirmed delivery date** check.

In the initialization parameters, you can specify tolerances for specific confirmation control keys, as well as opt to use the plant calendar.



Check confirmed Incoterms

/EBY/CL_PDOR_EVT_CHK_INCO

Checks the Incoterms 2 text given in the order confirmation against the Incoterms 2 text in the corresponding purchase order. If the two texts are different, a message is added to the message log.

In the initialization parameters, you can use the **Always verify** option to configure the check.

Initialization parameter	Description
X	**Verify even there was no value confirmed** The check is performed whether the Incoterms 2 text in the order confirmation has been entered or not.
blank	**Only verify when value confirmed** The check is performed only if the Incoterms 2 text in the order confirmation has been entered.

Check confirmed price

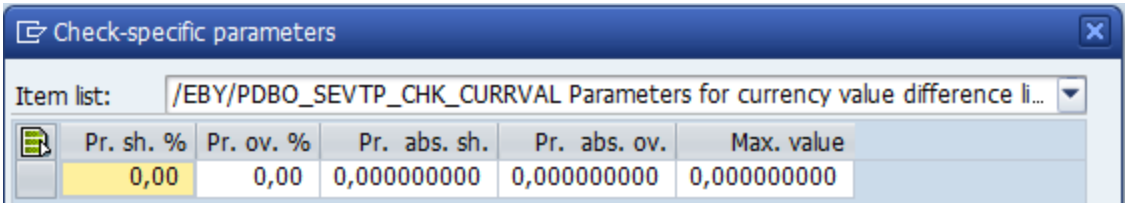
/EBY/CL_PDOR_EVT_CHK_PRICE

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.



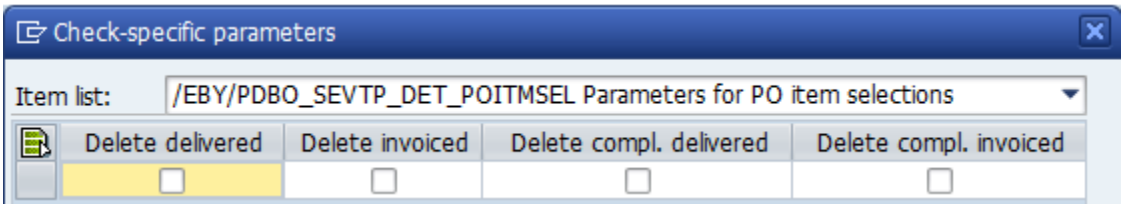
Check confirmed quantity

/EBY/CL_PDOR_EVT_CHK_QTY

Checks the quantity given in the order confirmation against the PO. If configured, the quantity conversion rules are taken into account.

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.

Check confirmed text

/EBY/CL_PDOR_EVT_CHK_TXT

Compares order confirmation line item texts with the texts of the corresponding purchase order line items and issues a message if the texts are not identical. Differences are displayed to the user in a popup window in the SAP GUI.

Check data from the action field

/EBY/CL_PDOR_EVT_CHK_ACTION

Generates an error message when an item is canceled.

Check confirmation is processed in the correct sequence

/EBY/CL_PDOR_EVT_CHK_ORDER

Verifies that order confirmation data is processed in the correct order. Unprocessed order confirmations are checked to determine if any of them contain the same PO item. If yes, the document with the lower PD number should be processed first. If the user tries to process the document that has a higher PD number, an error message is generated.

Determinations for Order Confirmations

The following standard determinations are available for Order Confirmations.

Purchase order line item assignment

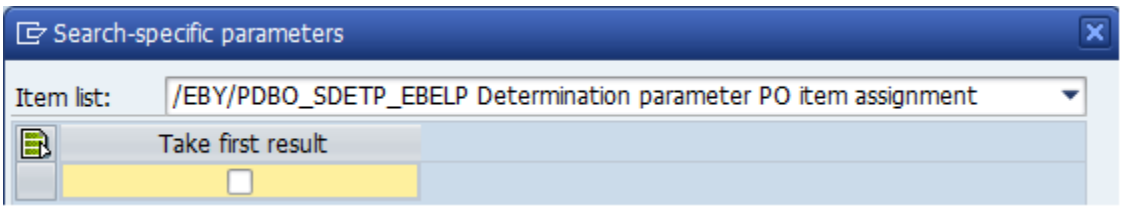
/EBY/CL_PDOR_DET_EBELP_INIT

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

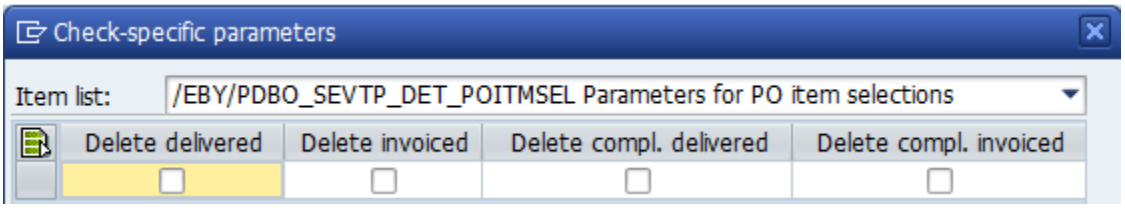
Search type	Search name
The vendor material number, line item number and quantity given on the order confirmation with the undelivered order quantity on the purchase order.	*/EBY/CL_PDOR_DET_EBELP_IDNLF*
The material number, line item number and quantity given on the order confirmation with the undelivered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

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 **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 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 Payment Approvals. See the *PROCESS DIRECTOR Configuration Guide* for more information.

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
SAP GUI	2	ITEMS
WA Document Detail	1	HEADER
WA Document Detail	2	ITEMS
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER

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

Type	Parameter	Description
	CS_BO_HDR	Data fields of a PD base object
	CS_OR_HDR	Order Response (OR) header
	CT_ITEMS	Order Response (OR) items

Payment Approvals (PR)

Checks for Payment Approvals

The following standard checks are available for Payment Approvals. 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 a user has deleted and recreated the payment proposal in SAP with the same identification.

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.
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 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 ReadSoft 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 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*
Item data	*/EBY/PDPR_MODEL*	*/EBY/PDPR_SITM_IF*	*CI_EBY_PDPR_SITM_DATA* *CI_EBY_PDPR_SITM_DISP*

BAdIs

BAdI implementation /EBY/PDPR_F110_SJOB

ReadSoft PROCESS DIRECTOR provides */EBY/PDPR_F110_SJOB* as an implementation of the standard SAP BAdI *FI_F110_SCHEDULE_JOB*. This implementation prevents the payment run from being carried out in `F110` if the payment proposal has not yet been approved in the PROCESS DIRECTOR workflow.

Note: */EBY/PDPR_F110_SJOB* can only be activated in client 000, but this automatically activates it in other clients as well.

Programs

/EBY/PDPR_IMPORT program

Use the `/EBY/PDPR_IMPORT` program to import payment proposals from SAP into ReadSoft 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 program

This program is only relevant if you have payment proposal documents that were created in a version of ReadSoft 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 ReadSoft PROCESS DIRECTOR 7.1 SP1, these connections were read whenever a document was displayed, which had a negative effect on performance. From ReadSoft 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)

Presets for Requisitions

The following standard presets are available for Requisitions. 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.

If you use this preset together with the **Preset accounting from item data** preset, place the **Preset accounting from previous item** preset after the **Preset accounting from item data** preset.

Preset accounting from item data

/EBY/CL_PDPO_PRESET_ACC_MAT

Automatically determines the G/L accounts before posting by calling the SAP `ME_ACCOUNT_ASSIGNMENT` function.

In the requisition, a value must be present in the **Account Assignment Category** field and in at least one of the following fields. PROCESS DIRECTOR searches the fields in the following order.

1. **Material**
2. **Material Group**
3. **Plant**

If you use this preset together with the **Preset accounting from previous item** preset, place the **Preset accounting from item data** preset before the **Preset account from previous item** preset.

Preset company code from purchasing org

/EBY/CL_PDPO_PRESET_COCO_PORG

Presets 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

Presets 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 preset uses the base unit of measure.

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 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 to the presets configuration. Create your own preset class based on the template and add this custom preset to the configuration.

Checks for Requisitions

The following standard checks are available for Requisitions. 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.

/EBY/CL_PDPO_EVT_CHK_ITEMCAT_B

Checks the limit amount.

Check Vendors returned from OCI

/EBY/CL_PDPO_EVT_CHK_OCI_VEND

Checks whether the OCI catalog returns more than one vendor.

Check SAP Purchase Order creation

/EBY/CL_PDPO_EVT_CHK_POST

Simulates the creation of the purchase order in SAP and displays 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 conditions and returns an error message if any of the conditions 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 the requisition contains 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
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.
Ask for decision PO or Requisition (Checks, Posting)	Determines if the popup that asks to confirm whether a purchase order or a requisition should be created is displayed or not. By default, this popup is not displayed.
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 *Activate outline agreements* in the *PROCESS DIRECTOR Configuration Guide*.
OCI field mapping (Other)	Provides parameters for mapping OCI data to ReadSoft PROCESS DIRECTOR: • **Mapping parameters for /EBY/CL_PDPO_EVT_OCI_UPDATE**: Map OCI catalog fields to PROCESS DIRECTOR fields. • **Mapping parameters OCI ISO code to internal code**: Map ISO codes in OCI data to internal measurement units. • **Additional parameters for /EBY/CL_PDPO_EVT_OCI_UPDATE texts**: Define which text object type ID is used for the long text from an OCI catalog.

Determinations for Requisitions

The following standard determinations are available for Requisitions.

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. ReadSoft PROCESS DIRECTOR enters this calculated delivery date for all the requisition's line items.

You specify the number of working days 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.

You can specify material groups and their corresponding G/L account numbers 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, such as company code or purchasing organization, based on the current processor of the document.

You can specify processors and their corresponding organization data 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 the following fields or a mapping table, 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 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. ReadSoft 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 Configuration Guide* for more information.

GUI area	Comp. type	Grid No.	Logical level
SAP document list	SAP GUI	1	HEADER
SAP Line items grid	SAP GUI	2	ITEMS
SAP Accounts grid	Additional grid 01	1	ITEM_ACCOUNTS
SAP Schedules grid	Additional grid 01	3	ITEM_SCHEDULES
SAP Conditions grid	Additional grid 01	4	ITEM_CONDITIONS
SAP Partners grid	Additional grid 01	5	PARTNERS
Web Application document detail header data	WA Document Detail	1	HEADER
Web Application document detail Line items grid	WA Document Detail	2	ITEMS

GUI area	Comp. type	Grid No.	Logical level
Web Application Accounts grid	WA Document Detail	3	ITEM_ACCOUNTS
Web Application document list	WA List Header	1	HEADER
Workflow status popup	Workflow status	1	HEADER
Archive log	Header data for archive log	1	HEADER

User exits / BAdIs

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

Type	Parameter	Description
	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

Type	Parameter	Description
	IT_PARTNERS	Requisition partner data
	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
	CT_SERVICES	Table Type for BAPIESLLC
	CT_SRVACCESSVALUES	Table Type for BAPIESKLC
	CT_PARTNERS	Table Type for BAPIEKKOP
	CT_ADDRDELIVERY	Table Type for BAPIMEPOADDRDELIVERY

Type	Parameter	Description
	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
	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

Type	Parameter	Description
	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
	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
	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

Type	Parameter	Description
	IT_ADDRDELIVERY	Table Type for BAPIMEPOADDRDELIVERY
	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
	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

Type	Parameter	Description
	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

Type	Parameter	Description
	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.

Down Payments

Check for Down Payments

The following standard check is available for Down Payments. You can also use the general checks that are available for all the process types.

Checks whether down payment can be posted

/EBY/CL_PDDP_EVT_CHK_BI_POST

Checks before posting by batch input. This check simulates posting by calling the F-47 transaction. This check is added for the **Check document** and **Initial processing** (document transfer from ReadSoft PROCESS DIRECTOR Accounts Payable to ReadSoft PROCESS DIRECTOR) actions.

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 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/PDDP_MODEL*	*/EBY/PDDP_SHDR_IF*	*CI_EBY_PDDP_SHDR_DATA* *CI_EBY_PDDP_SHDR_DISP*

View model customization

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

Comp. type	Grid No.	Logical level
SAP GUI	1	HEADER
WA Document Detail	1	HEADER
WA List Header	1	HEADER
Workflow status	1	HEADER
Header data for archive log	1	HEADER