ELO Suite for SAP ArchiveLink® (SAP NetWeaver® & SAP S/4HANA®)

ELO Connectivity Pack for SAP® ERP – Template Invoice

Table of contents

ELO Connectivity Pack for SAP® ERP – Template Invoice

Introduction

This documentation describes how to install and configure ELO Connectivity Pack for SAP® ERP – Template Invoice (Template Invoice for short in this document).

For read and transactional communication with SAP® ERP, Template Invoice exclusively uses features from ELO Connectivity Pack for SAP® ERP – Datatransfer, which is referred to as Datatransfer in this document. All information relating to Datatransfer about supported versions of SAP® ERP or ELO, the configuration (especially *searchHelps.config* and *functionModuleMappings.config*), and technical functionality or restrictions, applies equally to Template Invoice. Refer to the corresponding documentation in the ELO SupportWeb.

Template Invoice is essentially an extension of Business Solution ELO Invoice (referred to here as ELO Invoice) whose main function is to enable users to capture invoices in ELO and transfer them in the context of SAP® ERP.

Template Invoice is an implementation example with a range of preconfigured modules that can be easily reused in the implementation project. Due to the complexity of SAP® ERP systems and their customizing and programmatic extension options as well as the individuality of business processes, it may be necessary to make customer modifications to the basic configuration.

Requirements

Dependencies

As described in the Introduction, Template Invoice is based on ELO Invoice and uses Datatransfer functionalities for interface communication. You must have obtained both components and tested their functionality before the installation. The associated requirements and dependencies for SAP® ERP and ELO systems can be found in the corresponding documentation.

This implementation has been tested for the following ELO versions and components:

- ELOprofessional/ELOenterprise version 10.02.000 or higher
- ELO Invoice 1.08.000

> **Information**
>
> You need to take into account the dependencies of ELO Invoice in the aforementioned version. ELO Invoice updates and upgrades are not implemented at the same time as those of Template Invoice. In terms of the Template Invoice version, make sure you have the valid version of ELO Invoice, since the latter need not be the current version. New versions of ELO Invoice are evaluated and only implemented where necessary in Template Invoice during the next update or upgrade.

Security and licensing requirements

The security and licensing requirements for ELO Invoice and ELO Connectivity Pack for SAP® ERP apply. In particular, be aware of the possible implications for the SAP® ERP license when using Template Invoice with the Datatransfer functions as part of ELO Connectivity Pack for SAP® ERP.

Basics

The ability to process incoming invoices in SAP in the context of business transactions is dependent upon process integration between ELO and SAP® ERP, which requires advanced customizing in the relevant SAP modules.

> **Information**
>
> General rule: A request to transfer or post incoming invoices in SAP® ERP can only be executed if it can be manually posted with the corresponding SAP transaction. Conversely, the interface cannot implement a request if the invoice cannot be posted manually in SAP.
>
> Example: If customizing settings or master data are missing or incorrect (e.g. posting period is not open or invalid GL account), posting would fail regardless of whether it is done through the interface or manually.

Recommendation: Each posting variant in the target process should be manually verified in the respective SAP transaction before attempting to post through the interface.

Installation

Before installing Template Invoice, check the dependencies referred to in chapter Dependencies and install if necessary.

You will find the current installation file (ZIP folder) for Template Invoice on the ELO SupportWeb under:

Integration > ELO for SAP® ERP > ELO Suite for SAP ArchiveLink® > ELO Connectivity Pack for SAP® ERP > Download Templates

One component of this folder is an SAP transport that you need to import into the target system. All the *.eloinst* files must be installed in the ELO system in ascending order according to the numerical prefix in the file name.

SAP transport

Inform your customer or the SAP partner about the SAP transport to be installed and provide it to them.

> **Information**
>
> Inform your customers or SAP partner of the following:
>
> - The standard SAP transport routes must be used.
> - All developments and functions within the transport were developed in a separate SAP namespace, called */ELO/*.
>
> You may have to enable the *Ignore Invalid Component Version* option when importing the SAP transport.
>
> It is also possible that you will encounter warnings when importing the transport due to different release statuses (return code 8), which can be ignored. The transport object list also contains the class */ELO/CL_BADI_ACC_DOCUMENT* as an activated implementation of the Business Add-In (BAdI) *BADI_ACC_DOCUMENT*. More technical details about this class are described in chapter BAdIs. Any existing implementations for the same BAdI need to be checked for compatibility and modified if necessary.

Once the SAP transport has been imported successfully into the SAP systems, the other required packages can be installed.

ELO installation packages

The .eloinst files in the installation file must be installed in the following order:

1. 01_erp.sap.invoice_x.xx.xxx.eloinst
2. 91_custom_erp.sap.invoice_ x.xx.xxx.eloinst
3.

92_custom_sol.invoice_SAP_ x.xx.xxx.eloinst

The placeholder *x.xx.xxx* stands for the respective version number of Template Invoice.

> **Information**
>
> The *.eloinst* files also include an installation file, which is intended for the custom area of the package *sol.invoice* (ELO Invoice). When you install the package, all current settings in this area will be overwritten.
>
> This applies generally for reinstalling a package in the same administration path and is especially important for upgrades of Template Invoice or ELO Invoice.

Installation information

When installing Template Invoice, it is possible that the web apps for configuration files are not generated completely. You can tell that this is the case if there is no interface when you open the configuration.

1. To fix this, go to the ELO Application Server and open the ELOwf status report. You will see the *App Manager* entry in the menu bar on the right.
2. In the ELO App Manager, the *App status* indicates whether the web app installation was successful. If this is the case, the entry *Installed* is shown at the location. If it is set to *Archived*, you will have to enable/install, and refresh the web app.

Functionality

Template Invoice is an implementation example of an extension of ELO Invoice designed to enable bidirectional communication with an SAP ERP system. One component of the main extension is the direct integration of SAP search helps with form fields in ELO. This enables the user to execute search helps with dynamic filter criteria according to the entries in a form field directly in SAP and to output the results in the form field using a dynamic keyword list (DnyKWL).

In the context of processing incoming invoices, Template Invoice also shows how to load complex data structures of SAP objects into ELO and generate transactional data in SAP from ELO using the function module mapping (Datatransfer).

The search help and function module mappings provided with Template Invoice can be easily adapted to customer requirements by changing the parameterization. Reusable workflow components (subworkflows) make it easier to create custom ELO workflows.

The following subchapters provide an overview of the Template Invoice components and explain what modifications have to be made to ELO Invoice. You must have basic knowledge of ELO Invoice and the configuration and functionality of Datatransfer to understand the content.

Forms

All form customizations for Template Invoice are included as changes to the basic forms in the custom area of ELO Invoice. Following installation, you can check all changes in the form designer under *sol_invoice*.

> **Information**
>
> 
>
> You can recognize modified forms and form modules by the corresponding icon.
>
> Not all form modules provided with ELO Invoice are modified by Template Invoice. For example, the fields in the compact item line view are not linked to the SAP search help. It may be necessary to adapt the required form modules to meet the customer’s requirements.

Dynamic keyword lists/search helps

Most of the form customizations involve changing the dynamic keyword lists in existing form fields. Datatransfer enables direct integration with elementary SAP search helps. This eliminates the need to replicate master data or customizing objects, which are needed to enter data or validate a form, in an ELO database. Besides entailing less maintenance, this significantly enhances data quality since inconsistencies and time delays are prevented. Since SAP already provides a large number of elementary search helps for various objects, you do not often need to create custom search helps.

Dynamic keyword lists that address SAP search helps can be recognized by the prefix *erp.sap.invoice.ix.dynkwl*, e.g. *erp.sap.invoice.ix.dynkwl.CostCenter* for cost centers.

All search help mappings provided with Template Invoice can be viewed in the configuration file *searchHelps.config* under the administration path *Business Solutions // erp.sap.invoice // Configuration*. The result fields under *Output* or the filter fields used under *Search* usually allow you to determine the form field that is linked to the corresponding dynamic keyword list. Conversely, the search help mapping addressed by a form field can be recognized by the DynKWL script: The identifier following the aforementioned prefix is also the name of the mapping, e.g. the GL account in the item lines (map field *INVI_GL_ACCOUNT_NO*) is linked to the DynKWL *script erp.sap.invoice.ix.dynkwl.AccountGl*; the corresponding mapping is therefore called *AccountGl*.

Besides helping you to enter data in a field, search help mappings can transfer additional columns of the selected data set. For example, when selecting a tax code, the tax percentage returned by the search help is automatically written to another form field (see also search help mapping *TaxCodes*).

It is also possible to call search helps (or other DynKWL) dynamically from the form header script. For example, clicking *Load purchase order items* (*Invoice data* tab of the form, *Orders* area) executes the search help mapping *PurchaseOrderItems*, which reads the corresponding order items from SAP and transfers them to the invoice items in the form.

New form elements

To successfully post an invoice in SAP, you may require more than the data or available fields in ELO Invoice. For this reason, Template Invoice adds fields to the forms that are typically necessary for the most common posting variants of an incoming invoice in SAP. In addition, completely new form areas were created to make it easier to process incoming invoices in the SAP context.

The most important changes are summarized in the following list:

- Booking date
- *Credit note* check box for entering vendor credit notes
- *One-time account (CpD)* check box for posting to CpD accounts
- Payment block for open items
- Form area for payment terms: Payment terms can be loaded from SAP and, if required, modified in the form.
- Adding item lines for frequently used SAP fields, such as tax codes. Many of the item line extensions in Template Invoice are not enabled (hidden) in the standard software. These are intended for special incoming invoice variants and can be enabled if necessary (see Multiple account assignment of purchase order items (MM)).
- *SAP* form tab: Display error messages when posting and tracking the document flow of open items within SAP (parked/posted/balanced)

Additional functionalities/header script

All additional functionalities provided with Template Invoice are implemented in the integrated header script. It can be viewed under the following administration path: *Business Solutions // erp.sap.invoice // ELOwf Base // Webapp // erp.sap.invoice.forms.Invoice_JS*.

> **Please note**
>
> For update purposes, it is important that you do not modify this file or replace it with a file from the custom area. If you need to do customizations, you should integrate them in the form header script using an additional include.

The most important changes/additional functionalities are:

- Automatic enable/disable of the *Credit note* option depending on the selected document type. For details, see invoice.config – Accounting
- Exchange rates for foreign currencies are calculated based on the current rate and transferred with the corresponding search help for currencies in SAP. The booking date is used as the currency conversion date. Changing the booking date triggers a new background query in SAP, which automatically adjusts the exchange rate.
- Clicking *Load terms of payment* executes a search help query of the terms of payment key entered. The terms are transferred to the form where they can be changed by the user.
- Clicking *Load purchase order items* executes a search help query of the purchase order items entered. The loaded purchase order items are added to the item lines.

Workflow

Template Invoice provides a range of (sub)workflows that enable bidirectional communication between ELO and SAP based on Datatransfer. The workflow prefix is always *erp.sap.invoice*. The following workflows (can also be used in custom workflows) are included:

- erp.sap.invoice.Base: Basic workflow that maps the entire document capture process; starting with *automatic analysis* (e.g. ELO DocXtractor) followed by multi-stage approval/ release and error handling to posting and linking the SAP document. This workflow is a copy of *sol.invoice.base* of the ELO Invoice version used in each case; SAP data transfers are added where appropriate.
- erp.sap.invoice.Download: Subworkflow for loading additional information about the document. In the current version, the module is only used for automatically loading purchase order items in the case of an incoming invoice with reference to a given purchase order.
- erp.sap.invoice.Transfer: Subworkflow for transferring an invoice document to SAP. The system automatically distinguishes between the variants *document parking/direct posting* or *with/without order reference* and sends it to the respective function module mapping. You select between document parking and direct posting in the *invoice.config*, see also chapter invoice.config – General. In this case, you do not have to modify the workflow.
- erp.sap.invoice.Simulation: Subworkflow for the posting simulation or validation of invoices without an order reference. This workflow is optional and can be integrated if required; it is

not used in the standard software. You have to change the function module mapping that you want to call in the workflow directly.
- erp.sap.invoice.Link: Subworkflow for linking the documents generated in ELO with the corresponding documents in ELO. This adds the ELO document to the attachment list of the SAP document and determines the SAP object that is displayed when accessed from ELO Toolbox for SAP® ERP.
- erp.sap.invoice.ErrorHandling: Subworkflow for error handling. This workflow is called if errors occurred during transfer to SAP or processing within the SAP system (e.g. error messages in cases of inconsistent documents).

General comments

If you make changes to the basic workflow and, especially when reusing workflows, be aware that certain fields in the respective SORD are used to control the workflow. These are the following fields:

- ERP_STATE (metadata field): Status field for calling the function module mapping. The function module mapping writes the success or error status to this field. The possible status values in the respective version of Template Invoice can be taken from the subworkflows or the function module mapping.
- ERP_SAP_INVOICE_TRANSFERRED (map field): Determines whether and how the invoice was transferred to SAP (e.g. parked/posted). The workflow checks this field to ensure that the invoice is not transferred to SAP twice.
- WITH_PO_REFERENCE (map field): Specifies whether an invoice has a purchase order reference. Based on this field, the business transaction is controlled accordingly in the workflows (see posting in SAP with transactions *MIRO* vs. *FB60*).
- SAP_INV_DISTRIB (map field): Contains an aggregated view of the account assignment categories of all purchase order items (only if the invoice has an order reference). If there is at least one item with multiple account assignments (field is not empty), only the "invoice body" is transferred, see also Multiple account assignment of purchase order items (MM).

Transfer to SAP

Invoice data is transferred to the SAP system with the help of the Datatransfer function module mapping. An RFC-capable function module that can generate the transaction data is required for each business transaction to be posted in SAP. The SAP *Business Application Programming Interface (BAPI)* consists of a number of RFC-enabled function modules (referred to here as BAPIs) for posting various business transactions. The abbreviations *FI (Finance)* and *MM (Materials Management)* used in this documentation refer to the SAP modules of the same name:

- FI invoice: Invoice in financial accounting (without order reference, SAP transactions *FB01*, *FB60* etc.)
- MM invoice: Invoice in materials management (with order reference, SAP transaction *MIRO*)

BAPIs

Template Invoice uses the following BAPIs:

-

BAPI_INCOMINGINVOICE_PARK: Parks MM invoice documents, see SAP transaction *MIR7*
- BAPI_INCOMINGINVOICE_CREATE: Directly posts MM invoice documents, see SAP transaction *MIRO*
- BAPI_INCOMINGINVOICE_SAVE: Transfers the "body" of MM invoices containing items with multiple account assignments, see also Multiple account assignment of purchase order items (MM)
- BAPI_ACC_DOCUMENT_POST: Parks or directly posts FI invoice documents, see SAP transactions *FB01* or *FBV1*
- BAPI_ACC_DOCUMENT_CHECK: Validates or simulates FI invoices without generating transaction data in SAP

BAdIs

It can generally be assumed that the functionality and behavior of a BAPI is identical to its SAP counterpart (i.e. the corresponding SAP transaction). For example, in most cases, *BAPI_ACC_DOCUMENT_POST* returns the same result as transaction *FB01* with the same parameterization. This is not always the case, however. Some special cases, especially automatisms of an SAP transaction, are not easily mapped by the respective BAPI.

The SAP enhancement option *Business Add-Ins (BAdIs)* is used in cases such as these. Among other things, BAdIs allow you to enhance BAPIs using defined interfaces at specific times during internal processing. The enhancement is done in SAP (ABAP) by implementing a class with a predefined interface. You can refer to the relevant SAP BAPI documentation to find out whether or which BAdIs are available for a BAPI.

The Template Invoice SAP transport contains an active implementation of *BADI_ACC_DOCUMENT* as an enhancement of *BAPI_ACC_DOCUMENT_POST*. The enhancement class */ELO/CL_BADI_ACC_DOCUMENT* provides a solution for the following cases, which are not supported by the BAPI or not all versions of the BAPI:

- Post to assets (with subnumber and transaction type) with automatic determination of the asset reconciliation account
- Set an explicit transaction code in the document header
- Post accrual/deferral documents with reversal date (see SAP transaction *FBS1*)
- Cash discount net procedure

> **Information**
>
> Details of the above transactions and the corresponding parameterization are described in the respective subchapters of chapter Special business transactions.

The BAPI table *EXTENSION2* is transferred to the BAdI when called. This table must be correctly parameterized since the provided BAdI implementation only makes document changes on explicit instruction for security reasons. To prevent possible collisions with other BAdI implementations, the *STRUCTURE* field always contains a data element (existing in the ABAP Dictionary) in the namespace */ELO/* for all *EXTENSION2* lines transferred from ELO. The BAdI ignores all lines with different values in the *STRUCTURE* field. This ensures that the BAdI can only make document

changes when called by ELO but also minimizes the potential for conflicts with other BAdI implementations (if any) since it is bound to a reserved namespace. In any case, you need to check whether */ELO/CL_BADI_ACC_DOCUMENT* is functioning properly in the customer system (especially if other active BAdI implementations are in use).

The exact key of the data element is stored in the class constant */ELO/CL_BADI_ACC_DOCUMENT=>CO_PRM_ELOSTRUC*. We cannot guarantee that this key will not have to be changed for technical reasons in a future Template Invoice release. However, we do not expect the namespace to change.

The general structure of a line in *EXTENSION2* for the provided BAdI implementation is as follows:

- STRUCTURE: */ELO/EXTENSION2* (fixed value)
- VALUEPART1: Name of the parameter, e.g. *ASSET_MOVEMENT_TYPE*
- VALUEPART2: Number of the reference line according to BAPI, e.g. *3* (without line reference *empty*)
- VALUEPART3 and VALUEPART4: Value of the parameter; both fields are concatenated before processing

Use case and process example

This chapter describes how to enter an incoming invoice using the standard configuration of Template Invoice. The focus is on the ELO Invoice extensions, which enable users to capture all the invoice data necessary for successfully transferring the invoice to SAP. Knowledge of the standard functionalities and basics of ELO Invoice are required and will not be explained here.

The process example described here starts after a new invoice document has been stored. In many cases, ELO DocXtractor is already in use on the customer's system, and has already completed some or even all the necessary fields of the invoice. This process description does not explain how to configure and use ELO DocXtractor in the context of processing incoming invoices.

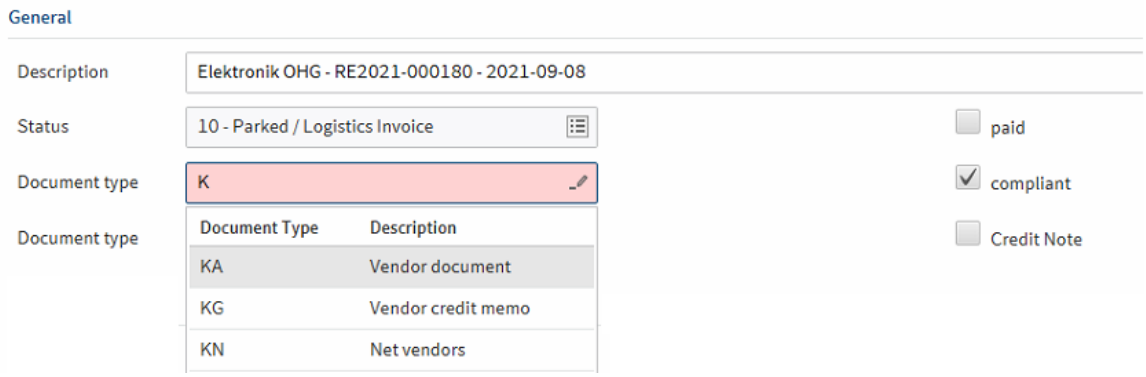
Automatic download

If header data already exists when the invoice document is stored, it is possible to retrieve additional invoice information from SAP in advance. To do so, you need to execute the subworkflow *erp.sap.invoice.Download.* This workflow checks whether the invoice has an order reference (map field *WITH_PO_REFERENCE*) and subsequently loads the items of the SAP purchase orders entered into the item data (function module mapping *getItemData*). If the SAP purchase orders are only known at the time of manual entry, the items can also be loaded via the form, see the Entering invoice data section.

Entering invoice data

In the form stored for the workflow item, the user can see all available information on the invoice as well as check, enter, or change information relevant to posting. The following section describes the different areas of the *Invoice data* form tab that are relevant in the context of Template Invoice.

General

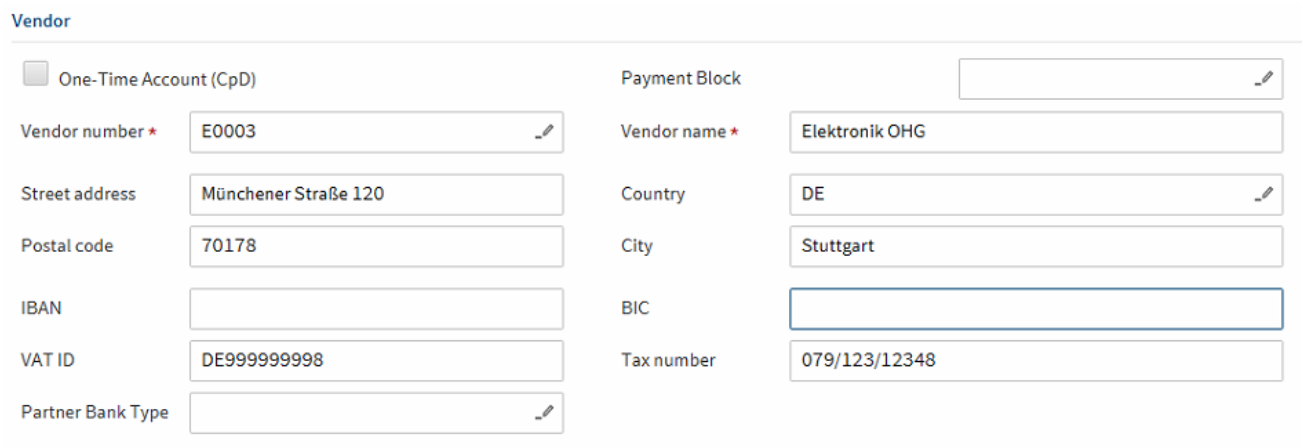


- The *Document type* also corresponds to the document type at header level of the SAP document. The field is linked to the corresponding SAP search help. If no document type or only the standard ELO Invoice document type *IN* has been entered, this value is automatically replaced with *KR* (vendor invoice in IDES systems) in the function module mapping so it can be transferred to SAP.
- In case of a *credit note*, the corresponding option must be enabled. Make sure that the document type is compatible with this option. To automatically determine the credit note code based on the document type, see chapter invoice.config – Accounting.

Company

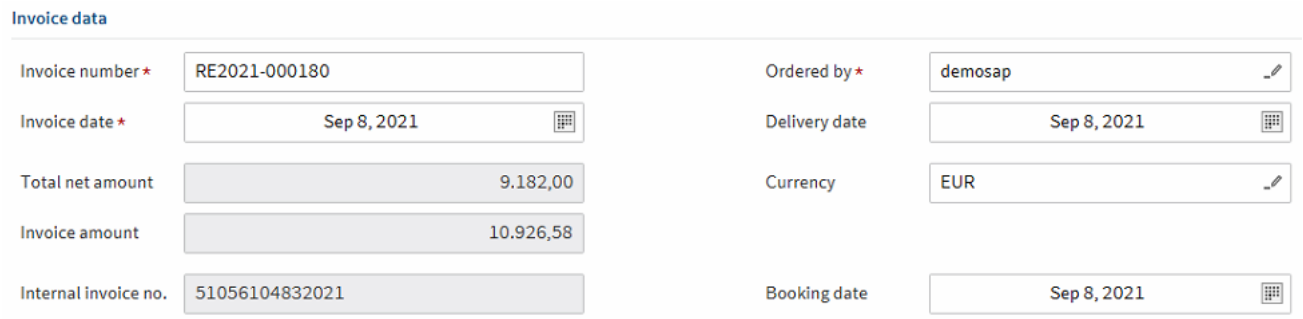
- The *Company* corresponds to the company code used to post the SAP document. The field is linked to the corresponding SAP search help. You have to make an entry in this field; the value is not automatically substituted during transfer to SAP.

Vendor



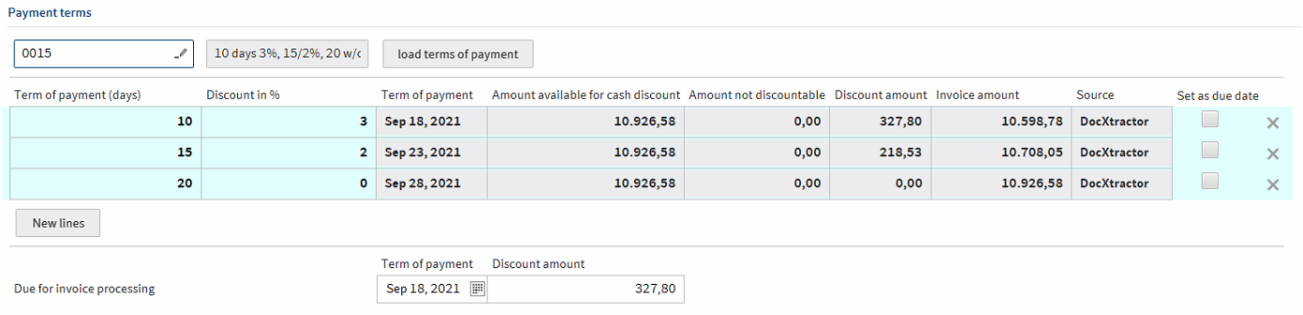
- The *Vendor number* corresponds to the number of the vendor account in SAP. The field is linked to the corresponding SAP search help. You have to make an entry in this field; the value is not automatically substituted during transfer to SAP. When you select a vendor in the dynamic keyword list, the name and address data are taken from the vendor master data in the form.
- If using a CpD account, the option *One-Time Account (CpD)* must be enabled. This ensures that the search help for the *vendor number* only displays CpD accounts.
- During posting, a *payment block* is set on the open items in the SAP document. The field is linked to the corresponding SAP search help.

Invoice data



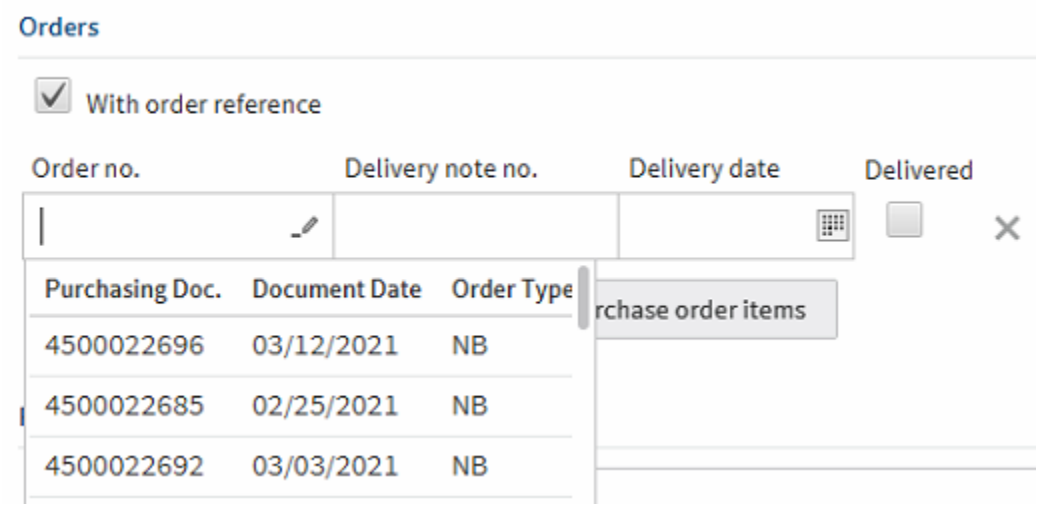
- The *Invoice date* corresponds to the *document date* and the *baseline date for payment* in the SAP document.
- The *Booking date* is also the booking date for the SAP document.
- The *Currency* is the same as the *document currency* of the entire SAP document including its items. The field is linked to the corresponding SAP search help. If you select a foreign currency, the respective exchange rate is updated in the ELO Invoice form field provided. The conversion date for the SAP exchange rate is the *booking date*. The SAP table *TCURR* is used for the currency conversion.
- The *Internal invoice number* is automatically updated by the function module mapping after successful posting and is used for internal processing.

Payment terms



- The input field for the *payment terms* is linked to the corresponding SAP search help.
- If you click the button *Load terms of payment*, you can insert the respective terms into the table below. Technically, the payment terms are retrieved in the background on execution of the search help mapping *PaymentTermsLines*.
- If there are entries in the table, these are transferred to SAP as payment terms for the open item. These take priority over the customized terms of payment key. Only the first three (highlighted in color) lines in SAP are taken into account. Manually entered or changed payment terms must comply with SAP requirements (e.g. days in ascending order, percentages in descending order, etc.).
- If there are no entries in the table, SAP automatically takes the respective payment terms from the customized terms of payment key.

Purchase orders



- In the case of MM invoices, the *With order reference* option must be enabled.
- You can enter the purchase order numbers in the table below or select them using the SAP search help.
- The *Load purchase order items* button loads the purchase order items from SAP and inserts them into the item table (*Item data* tab). Technically, the purchase order items are retrieved in the background on execution of the search help mapping *PurchaseOrderItems*.

Entering item data

Once you have entered the invoice or header data, the invoice items are entered in the *Item data* form tab. Each invoice item must contain sufficient information for posting in SAP. This usually implies a *GL account*, a *controlling object* (e.g. cost center), *quantity*, *unit price*, and a *tax code*. All fields with a reference to an SAP object are linked to a corresponding search help. Which fields in which constellations are actually mandatory in the customer system depends on its customizing requirements and, if applicable, custom validation and substitution rules. The required posting rate for each scenario must be approved by the responsible accounting staff and technically validated.

When a tax code is selected, the corresponding tax rate is automatically updated for the item. Based on the quantity, unit price, and tax rate, it is possible to automatically calculate the other amount fields in the item lines (e.g. tax amount, gross amount, etc.).

> **Information**
>
> Automatic calculation or retroactive calculation of amount fields is part of ELO Invoice; details and possible restrictions can be found in the corresponding documentation.

In addition, the following technical provisions apply for entering item data:

- When you select a tax account, the line item is posted as direct tax (for details, see chapter Direct tax (FI)).
- The *Item number* field corresponds to the *material number* in SAP. This field is linked to a corresponding search help.
- In the case of tax codes with complex tax transactions (e.g. VAT within the EU), the tax lines must be generated in SAP. For details, see chapter Complex tax transactions (FI). If you select these types of tax codes, the *Calculate tax in SAP* code, which cannot be changed in the form, is automatically updated by SAP for each item.

Entering footer data

As provided in ELO Invoice, you can enter surcharges in the *Footer data* form tab. Remember to consider the requirements of the respective SAP system in terms of account assignment data, as explained in the Purchase orders section. A new document is created with each surcharge, and where applicable, a tax line in the SAP document. Template Invoice does not provide an option for posting discounts in SAP. Depending on the SAP customizing, it would be possible to allow for negative surcharges, but this solution would have to be validated in the respective system.

The non-modifiable field *Calculate tax in SAP* in the center of the footer data is enabled as soon as the code of the same name is enabled in at least one invoice, surcharge, or discount item. This code is used as an auxiliary field for internal processing.

Release and approval

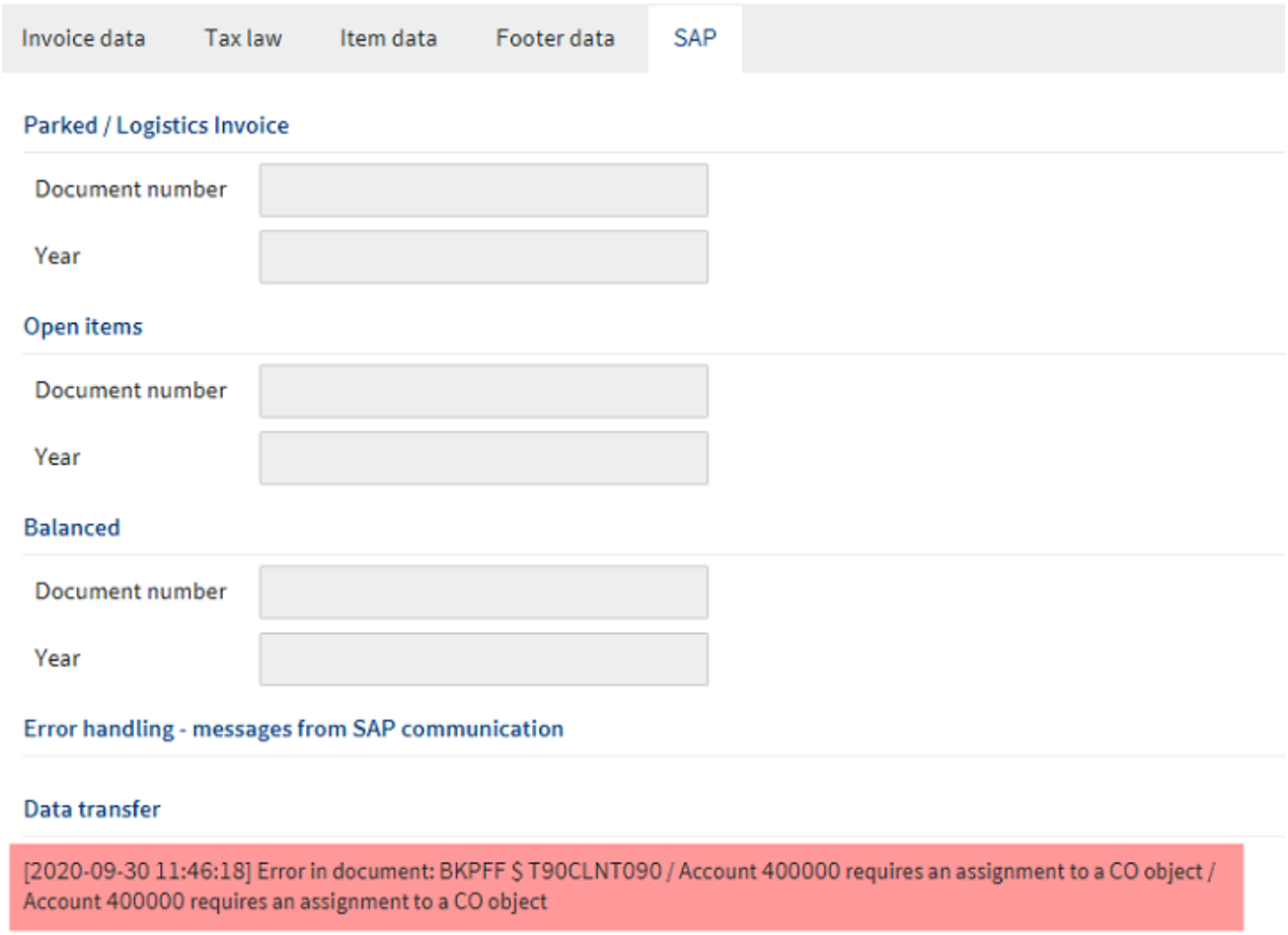
Once the document has been entered, validated, and released, a multi-step approval process takes place in accordance with the basic ELO Invoice workflow. During the individual steps, the respective users are given role-based access to the document data and can make changes if necessary.

> **Information**
>
> Template Invoice does not extend every (role-based) form module in ELO Invoice. For example, there may be fields with a reference to SAP that are linked to a corresponding search help. Depending on which form modules are going to be used in the customer's process, it is possible that some form modules will need to be changed. The form modules modified by Template Invoice are used as a template.

Transfer to SAP

After all necessary approval steps have been completed, the invoice is transferred to SAP (ERP export). To do so, you need to execute the workflow *erp.sap.invoice.Transfer*. For technical basics of the transfer process, see chapter Transfer to SAP.

ErrorHandling



If errors occurred during transmission, the basic workflow branches off to the *ErrorHandling*. With the corresponding workflow item, the user has access to the form and can fix errors. All errors that occurred during transmission are recorded in the *SAP* form tab. In this context, we have to distinguish between technical errors and business errors:

- Technical errors are error messages that are returned by the called function module in SAP during document validation, e.g. *posting period is not open*. BAPIs with the same cause of error also output the same error messages in the corresponding SAP transaction (e.g. *FB60*).
- Technical errors normally occur during processing in ELO and typically mean the document is not transferred to SAP. You can usually recognize these kinds of errors by a stack trace at the beginning of the error message. Common causes of errors:
- Connection cannot be established (e.g. incorrect/insufficient configuration of the connection settings in Datatransfer or a network error)
- Syntax error in function module mapping, especially with Handlebars expressions
- Runtime errors in a script (e.g. preprocessor in function module mapping)
- Special case: Runtime errors during execution in SAP. This can occur especially with custom function modules and enhancement implementations.

Once the cause of the error has been eliminated, the document can be transferred to SAP again.

Linking

As soon as the document has been successfully transferred to SAP, the link is established via the subworkflow *erp.sap.invoice.Link*. Once it is successfully linked, the ELO document can be found in the attachment list of the SAP document.

Tracking the posted document

The document number and fiscal year of the document posted in SAP is recorded in the *SAP* form tab. The document flow within SAP is periodically tracked by the ELOas rule *erp.sap.invoice.as.DocumentState* and updated in ELO. The individual document number categories are:

- Parked/Logistics Invoice: All documents generated out of ELO (only parked) or all MM documents
- Open items: Either FI documents posted directly from ELO (not parked) or the follow-up document of an MM document or parked FI document
- Balanced: The clearing document for the open item (e.g. through payment in the payment program)

To use document tracking, the option *Open item management* in the respective company code customizing must be enabled.

ELO configuration

Template Invoice offers a wide range of configuration options to adapt the functionality to individual needs. All search help and function module mappings provided as standard are based on Datatransfer and are therefore not explained in detail. This chapter focuses on the configuration options of *invoice.config.* Selected search help and function module mappings of particular importance are also explained in parts.

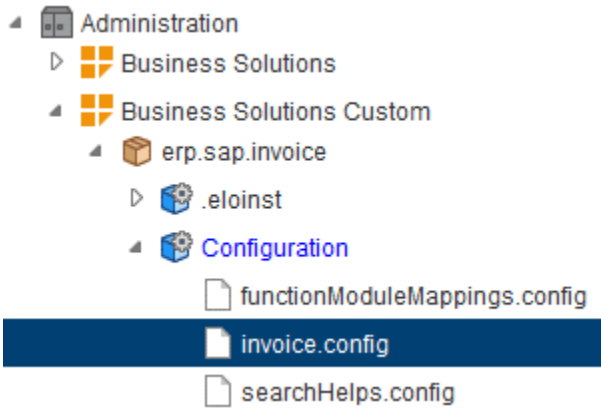
Please note

You should only make changes to the configuration of Template Invoice in the *Business Solutions Custom* folder to ensure you can update the system in the future. All files in the base folder may be deleted or overwritten when an update is installed.

Information

During installation, you may not be able to open the configuration interface of configuration files, see chapter Installation information.

Invoice.config

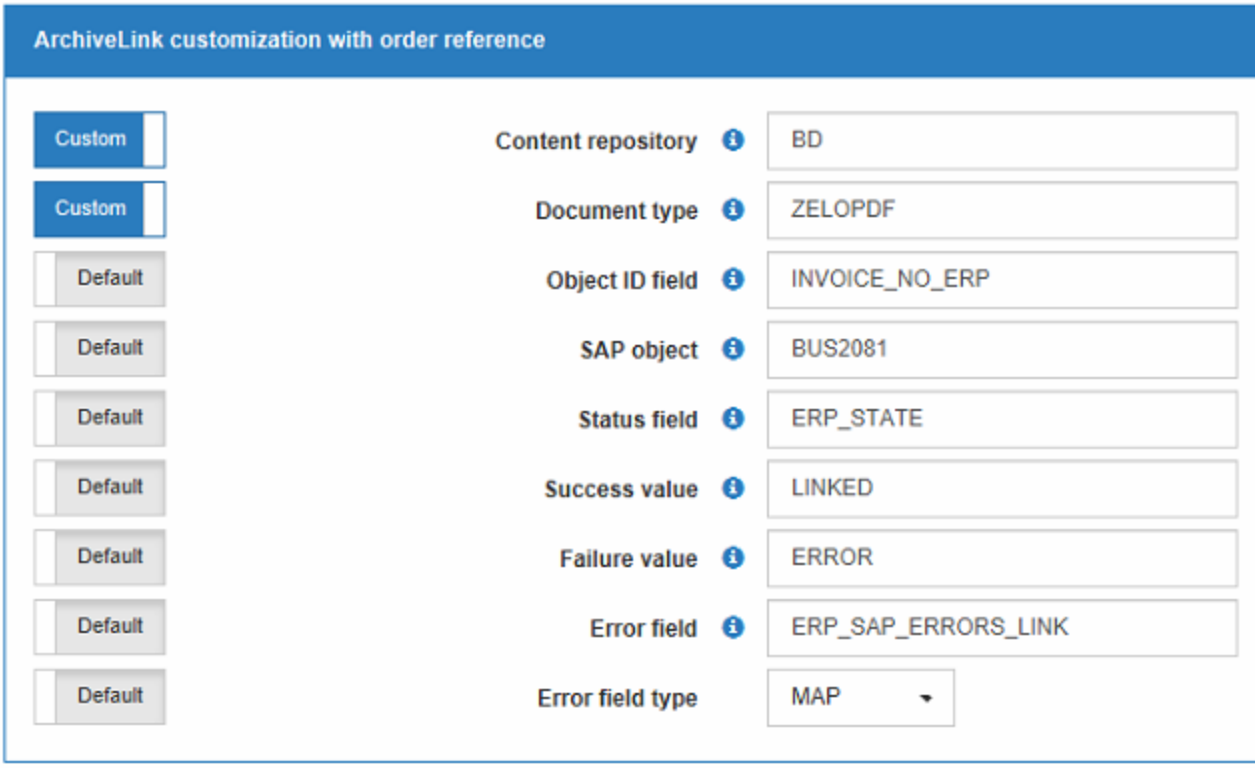


The configuration interface has several tabs: *General*, *Accounting*, *Status Query*, and *Posting Lines* (in expert mode).

invoice.config – General

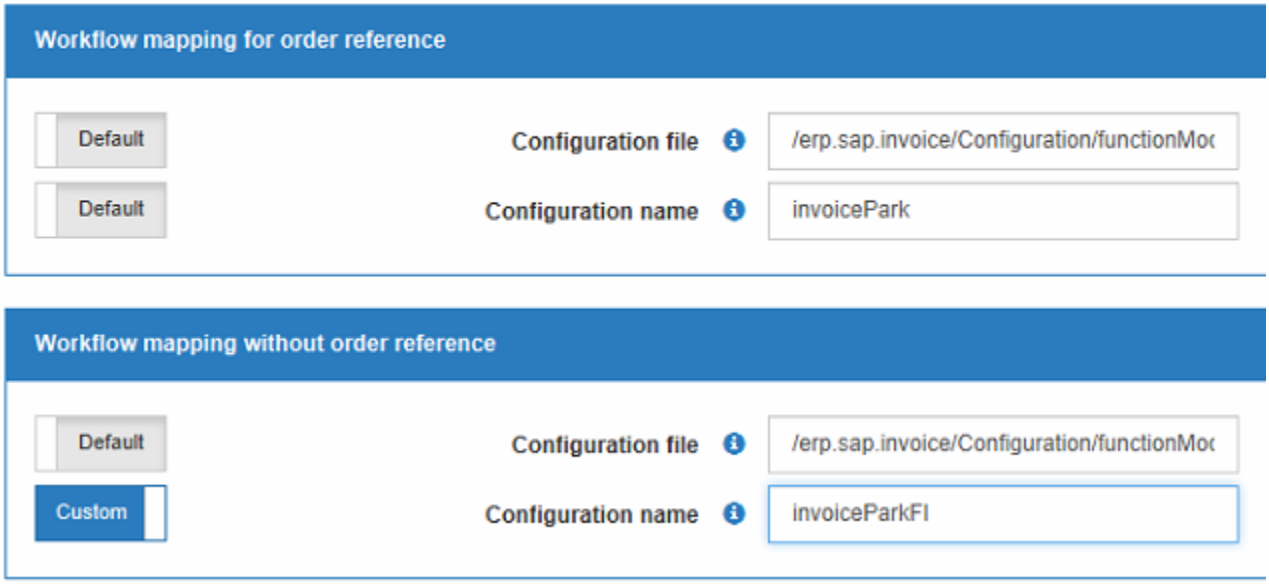
You always have to make settings in the *General* tab. This tab primarily contains the settings for SAP ArchiveLink for each type of invoice object in SAP, i.e. MM invoices with order reference and FI invoices without order reference. The following figure shows the fields which you should check and modify if necessary:

- Content repository
- Document type



You do not usually need to modify the default values. This is only required in special cases. It is especially important to note that the workflows provided with Template Invoice have dependencies with these settings. For example, the *Link* subworkflow expects the default success and error values in the default status field. If changes are made, the workflow must be checked and modified if required.

The configuration for invoices *without order reference* is identical.



Under *Workflow mapping for purchase order reference* or *Workflow mapping without purchase order reference*, you define which function module mapping is to be executed for posting FI or MM

invoices. Specifically, the *Transfer* subworkflow reads the mappings to be executed each time it is executed from this configuration. In the standard version, you can choose between parking and direct posting:

- invoicePark: Park MM invoice
- invoicePost: Directly post MM invoice
- invoiceParkFI: Park FI invoice
- invoicePostFI: Directly post FI invoice

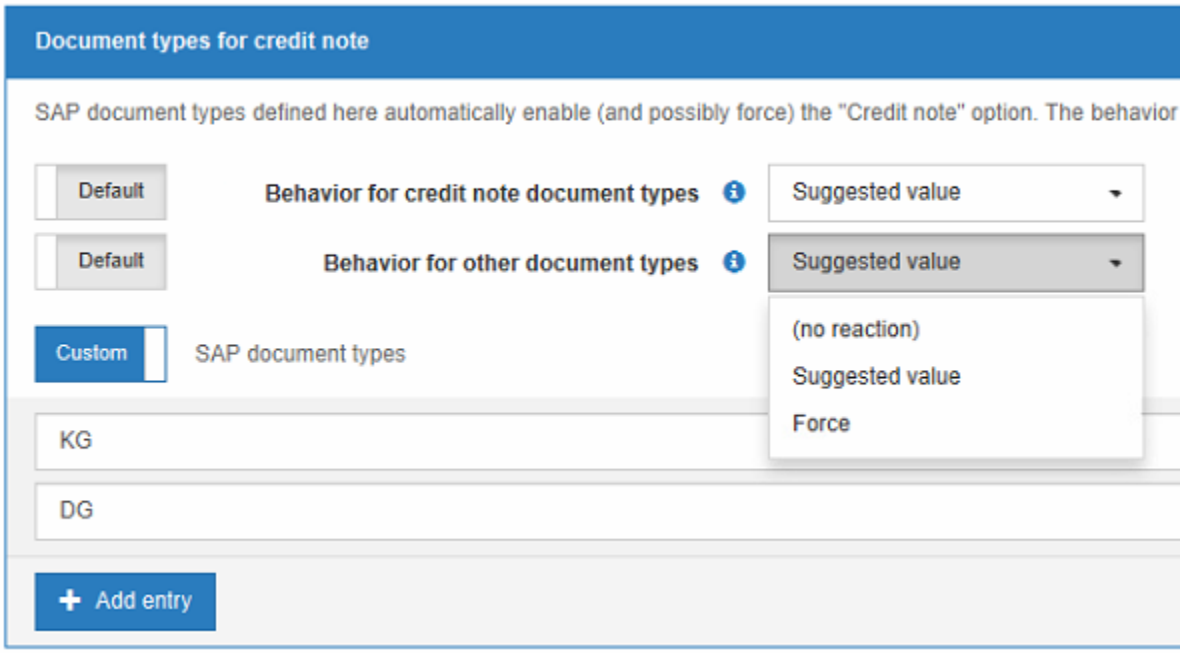
In addition, you can enter any (specially created in *functionModuleMappings.config*) mapping name. You can even interchange the entire configuration file if required.

The special case of transferring an invoice body or the mapping provided for it should not be configured at this point. For more details, see Multiple account assignment of purchase order items (MM).

invoice.config – Accounting

The *Accounting* tab contains settings that influence the behavior of the form and the accounting logic in SAP.

Document types for credit note



This area contains the options for handling credit note document types in SAP. Here, you can automatically enable or disable the *Credit note* code in the form depending on the selected document type.

The first step is to identify all SAP document types intended for credit notes and enter them in the list. In the next step, you define the behavior of the form when you select a document type that is or is not in the list.

When you select a document type, the setting *Behavior for credit note document types* takes effect:

- (no reaction): The *Credit note* code is not changed.
- Suggested value: The *Credit note* code is enabled but can be disabled by the user.
- Force: The *Credit note* code is enabled and cannot be disabled by the user.

When you select a document type that is not in the list, the setting *Behavior for other document types* takes effect:

- (no reaction): The *Credit note* code is not changed.
- Suggested value: The *Credit note* code is disabled but can be enabled by the user.
- Force: The *Credit note* code is disabled and cannot be enabled by the user.

Technically, the *Force* option sets the corresponding check box on the form to *read-only*. However, this setting is not permanent. If the form is reloaded, the check box can be changed again.

Cash discount: Automatic net procedure

If this option is enabled, the cash discount net procedure is automatically applied if the selected document type is a net document type. For details about this type of transaction, see Cash discount net procedure (FI).

invoice.config – Status Query

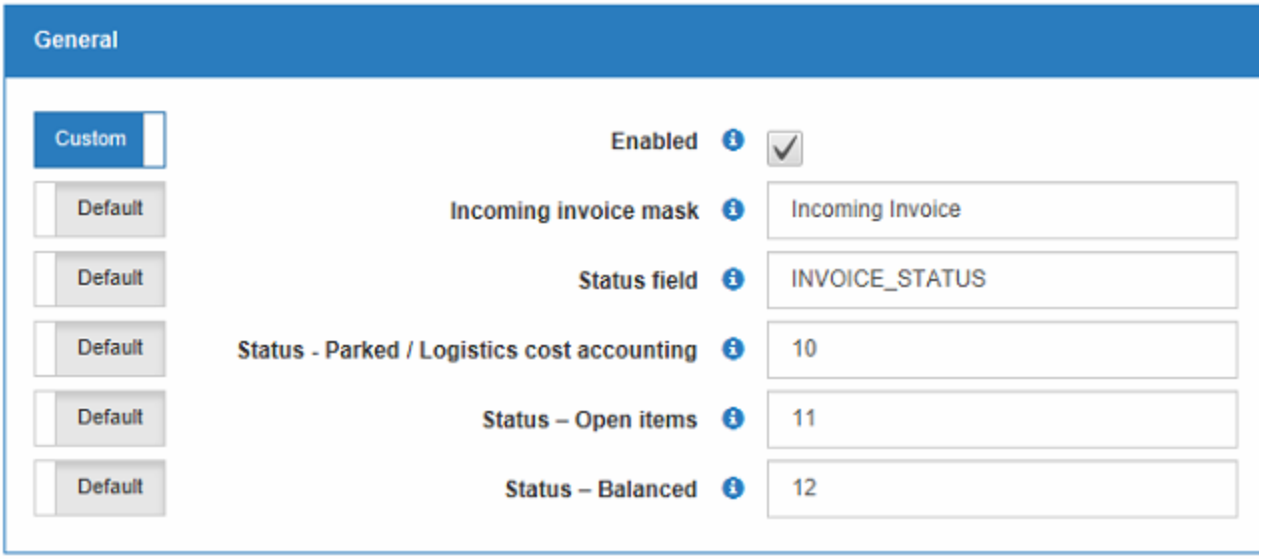
In addition to the status values provided in ELO Invoice, Template Invoice includes the following statuses:

- 10: Parked FI invoice or parked or directly posted MM invoice
- 11 (Open items): Directly posted FI document or follow-up FI document of an MM document
- 12: Clearing document for the open item

The aforementioned values have been added to the corresponding dynamic keyword list for status values in the custom area of *sol.invoice*. The status values are written in the subworkflow *erp.sap.invoice.Link* and in the ELOas rule *DocumentState*.

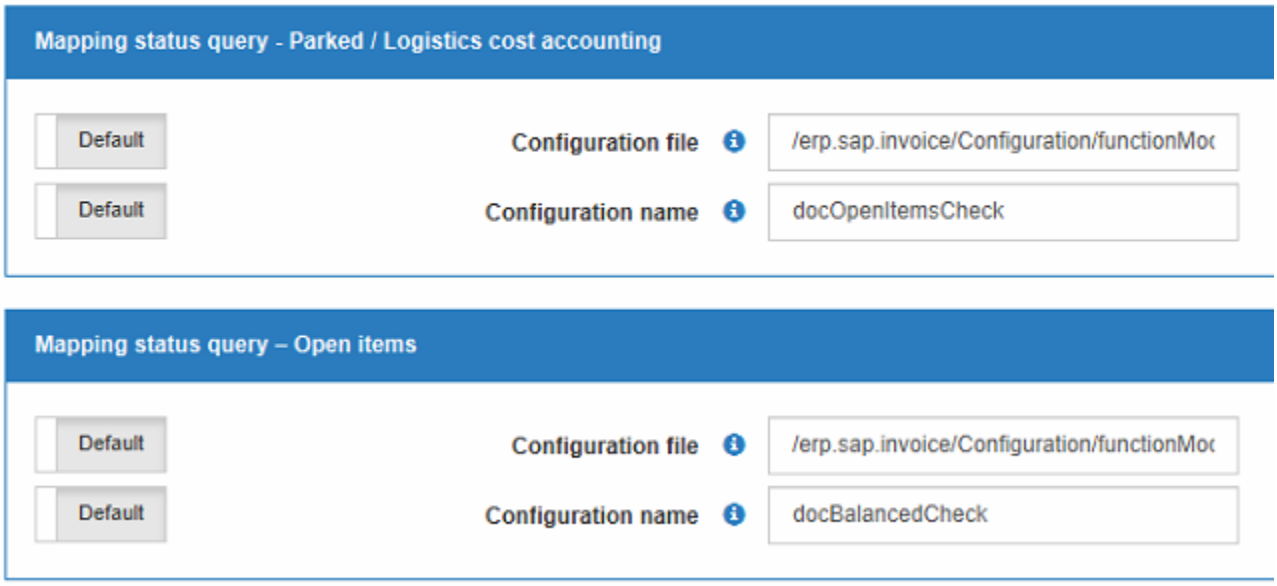
The *Status Query* tab is where you configure tracking of the invoice document flow in SAP described in chapter Use case and process example.

General



If you want to track the document status, you need to enable this option under *General*. You do not usually need to modify the other fields in this area. However, if you use a form for incoming invoices that differs from one of the default forms in ELO Invoice, this form needs to be entered here along with the corresponding status field. If the status values are changed, you may need to make these changes known to ELO Invoice.

Mapping status query



The function module mapping used for tracking parked FI and MM documents or open items in FI can be changed in the two *Mapping status query ... areas* if required. The corresponding ELOas rule reads the respective mapping dynamically from this configuration.

Check status change

The two *Check status change...* areas is where you specify the conditions under which the status change takes place. There are two possible status changes:

- Parked/Logistics cost accounting (10) ➔ Open items (11)
- Open items (11) ➔ Balanced (12)

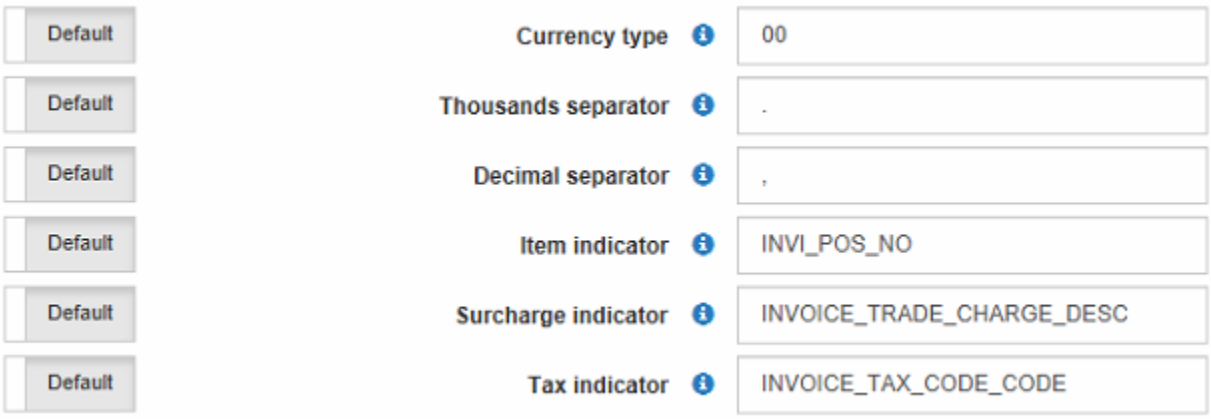
For both status changes, you can specify which value a field must contain in order for the status change to work. If you enable *not empty*, the field only needs to exist and contain a value.

The mappings used in the basic configuration write the document number of the open item and later of the adjustment posting to the SORD. Since the document numbers are recorded in different fields, you only need to check that the field is *not empty* for the status change to be executed.

invoice.config – Posting Lines

You only see the *Posting Lines* tab after you have enabled the expert mode. This is where you configure the preprocessor for FI postings, known as the *posting line script*. The posting line script is only used for FI invoices, i.e. the script is the preprocessor for the mappings *invoiceParkFI* and *invoicePostFI*, which both address the SAP function module *BAPI_ACC_DOCUMENT_POST*. For details about the posting line script, see 'invoicePostFI' mapping.

Preprocessing function posting lines

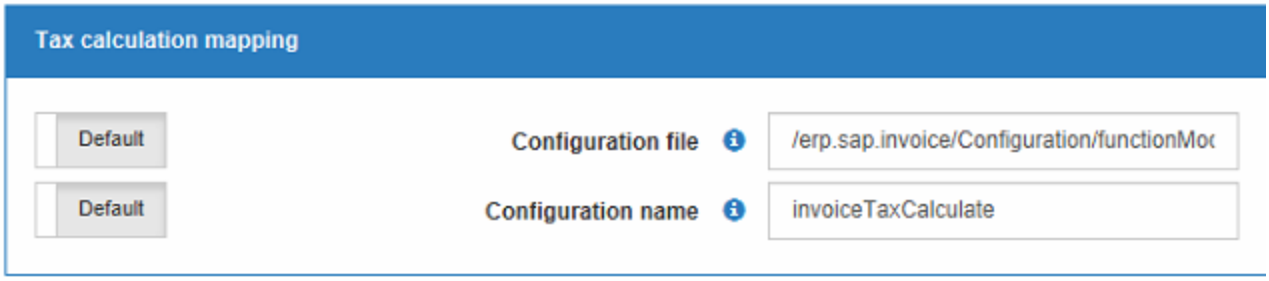


This section contains a description of the individual fields in the *Preprocessing function posting lines* area. All referenced tables or structures and their fields refer to the interface of the SAP function module (BAPI) mentioned above.

- Currency type: SAP currency type for all amounts, default: 00 (document currency); is set in the BAPI as a fixed value in each currency line (CURRENCYAMOUNT-CURR_TYPE).
- Thousands separator: Thousands separator used to interpret amount fields
- Decimal separator: Decimal separator used to interpret amount fields
- Item indicator: Indicator field * (map field) for item lines
- Surcharge indicator: Indicator field * (map field) for item lines
- Tax indicator: Indicator field * (map field) for item lines * Indicator fields control the loop over the lines a table. As soon as the indicator field in a table row is empty, the loop stops and the

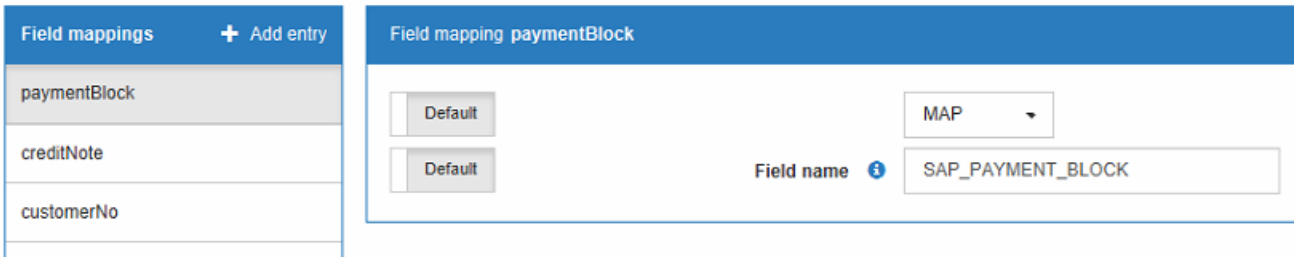
table is no longer processed. For each table, you should therefore select one field which is filled in each line to be processed.

Tax calculation mapping



As described in Use case and process example, a separate tax calculation sometimes needs to be done in SAP (for technical details, see Complex tax transactions (FI)). If the corresponding code is set in the footer of the document, the posting line script executes the tax calculation mapping defined in the configuration before it is transferred to the BAPI and accepts the returned tax lines. You can change the respective mapping and/or the corresponding configuration file at this point.

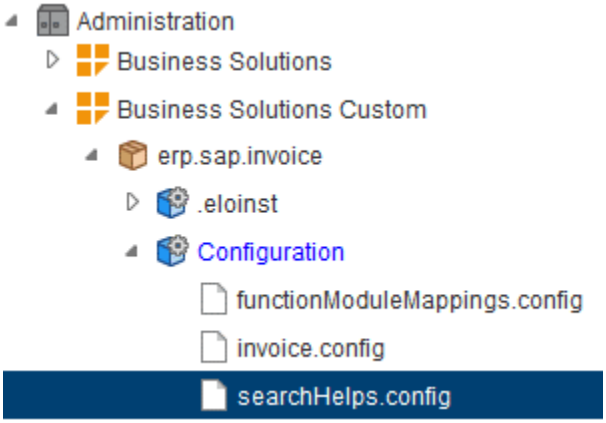
Field mappings



The *Field mappings* area acts as an abstraction layer to ensure that you do not need to hard code fields or field values in the posting line script. The field list on the left side contains the internal identifiers for each field as they are expected and read by the posting line script. Once you have selected an identifier, the map, metadata, or SORD field is mapped on the right side.

This way, the mapped fields of the basic configuration are easily interchangeable. It is also possible to add new field mappings with new identifiers. However, this only makes sense if associated with a modification of the posting line script. Although the new identifier and the corresponding field value are made available to the basic script, it will not be accessed since the identifier is not known to it.

Search help mapping



You are required to have basic knowledge of the search help mapping and its configuration as well as how to use search helps as dynamic keyword lists.

Search helps in Template Invoice serve the following purposes:

- To help the user complete form fields by displaying a filtered value list with additional information about the object (identical to search helps in SAP GUI)
- To transfer additional information about the selected list object (e.g. vendor address)

In many cases, the function module mapping is not required to retrieve data from SAP in the form context. Instead, all fields in an SAP search help output can be written to a suitable field in the form. This is beneficial for performance since the actual list values and the associated metadata only need to be transferred to ELO. In addition, you have access to data sources within all the elementary search helps available in the SAP system and not just RFC function modules.

All search help mappings provided can be viewed in the configuration file *searchHelps.config* under the administration path illustrated above. The following section only contains a few special search helps and general remarks.

'TaxCodes' mapping

The SAP search help */ELO/SH_TAX_CODES* provides tax codes (if necessary, also broken down into levels according to *tax jurisdiction codes*) and additional information necessary for the generation of the tax lines.

Complex/multi-level tax codes (e.g. for the reverse charge system within the EU) are always assigned a tax rate of 0.00% and a corresponding code (for details see Complex tax transactions (FI)).

When selecting a tax code, the following additional information is transferred (for specific field names, see Configuration):

- Tax rate (or 0.00 %, see above)
- Tax account
- Tax transaction key
-

Tax condition type
- Code for complex tax transactions, see above

'CurrencyRates' mapping

If invoices are issued in a foreign currency, it is usually preferable to have the amounts displayed and updated in local currency in an ELO program, e.g. for simple plausibility checks of the amounts or for use in subsequent processes within ELO (e.g. ELO Invoice dashboard). Since the SAP system requires the respective valid exchange rates for posting in foreign currencies anyway, Template Invoice replaces the exchange rate determination of ELO Invoice with a corresponding search help. The SAP search help */ELO/SH_CURRENCY* determines the exchange rate between a local currency and all foreign currencies in the respective hit list. The exchange rate of the selected foreign currency is updated in the ELO Invoice form field provided.

The following search help parameters are especially important:

- HOME_CUR: Local currency (usually configured in ELO Invoice)
- WAERS: Key of (foreign) currency
- FXDATE: Exchange rate date for currency conversion
- FXTYPE: Exchange rate type for currency conversion
- INVERSE_RATE: Indicator for reversing the SAP exchange rates. If enabled, the inverse value is determined for all foreign currency rates determined in SAP before output. This should be enabled for compatibility with ELO Invoice.

You can also store an ISO code in SAP for each currency key. It does not have to be identical. ISO codes are ignored in the basic configuration, which means that only the SAP currency key is used for the form entry and posting in SAP. Besides the currency field in ELO Invoice (*INVOICE_CURRENCY_CODE*), an additional map field (*IX_MAP_ERP_SAP_CURRENCY*) is provided for extension purposes. This additional field provides an option to distinguish between the currency key that ELO Invoice processes and the currency key used for posting in SAP.

Since a search help field can only be written back once due to technical restrictions (but not multiple times into different SORD fields), the search help contains the two fields *WAERS_COPY* and *ISOCD_COPY*, which contain the same value as the fields *WAERS* and *ISOCD*, respectively. This ensures the greatest possible flexibility during configuration.

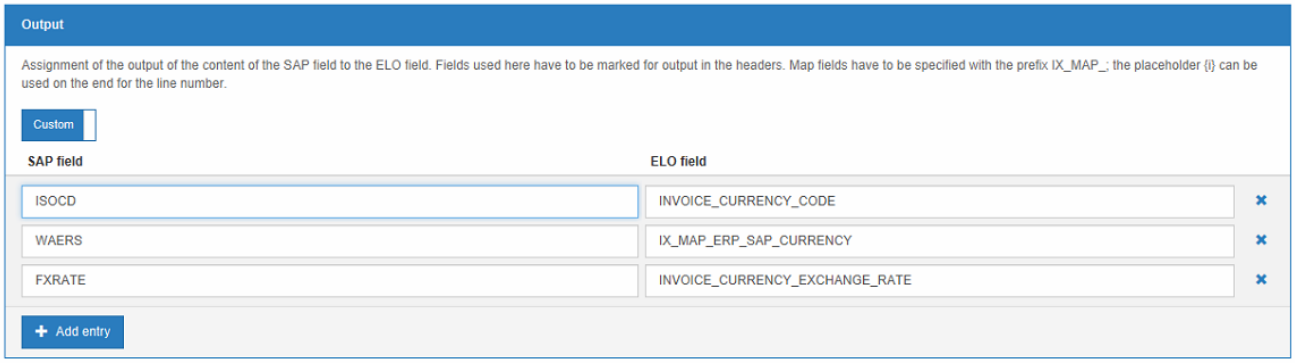
If you want to change the behavior, you must make the following settings according to the scenario:

Scenario 1: Update ISO codes/post with SAP currency keys

In this scenario, we want the ISO codes in the standard field for the document currency *(INVOICE_CURRENCY_CODE)* to be updated, e.g. to align deviating currency keys from different sources. However, we want to use the SAP currency key for posting.

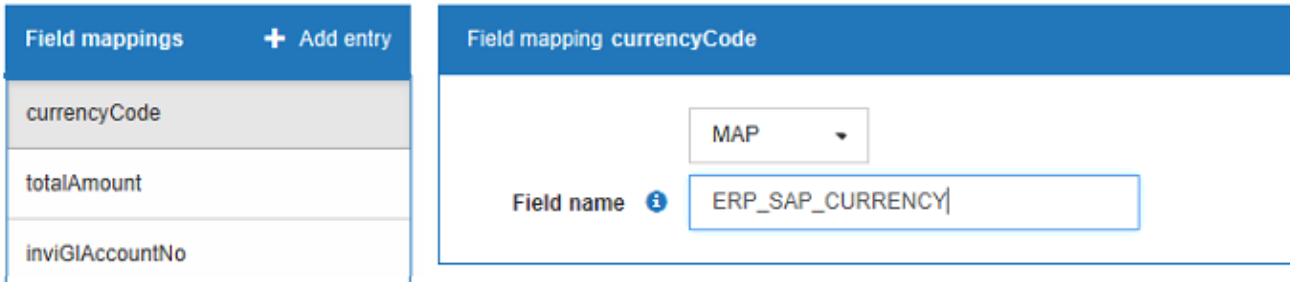
Required settings:

1 Search help output CurrencyRates



The ISO code is written to the standard field; the SAP currency key is written to the additional field.

2 Field mappings posting line script (without order reference)



In the field mappings for the posting line script (*invoice.config*/expert mode), the additional field has to be assigned to the script field *currencyCode*.

3 Function module mapping

You need to check whether the additional field is transferred as a currency key for all posting-relevant function module mappings. In the case of invoices without an order reference, each line item is accessed using Handlebars syntax:

```
{{{preObj.currencyamount.{I}.currency}}}
```

In the case of invoices with an order reference, the map field *ERP_SAP_CURRENCY* is assigned directly.

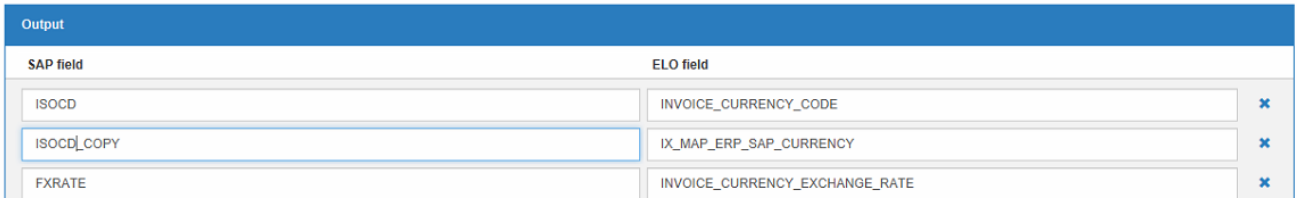
Fields for ISO codes should not be transferred in this scenario, e.g. you need to remove the check mark next to *Transfer* for the field *CURRENCYAMOUNT-CURRENCY_ISO* for all relevant mappings for *BAPI_ACC_DOCUMENT_POST*.

Scenario 2: Update and post with ISO codes

In this scenario, we want to update the document currency in the context of ELO Invoice and post the invoice in SAP based on ISO codes.

Required settings:

1 Search help output CurrencyRates



The ISO code is written to the standard field and the additional field.

2 Function module mapping

You must disable the option to transfer SAP currency keys in all relevant mappings, e.g. the field *CURRENCYAMOUNT-CURRENCY* in all relevant mappings for *BAPI_ACC_DOCUMENT_POST*. Instead, only corresponding ISO code fields should be transferred, e.g. *CURRENCYAMOUNT-CURRENCY_ISO*. For more information about parameterization, see scenario 1.

You should generally check the specific requirements in the customer system regarding the transfer of ISO codes. There is no guarantee that every SAP system supports the transmission of ISO currency codes alone.

The settings described above should only be made if the SAP currency key and the ISO codes assigned to it are different. Aligning them in SAP is another solution and would eliminate the need for additional settings in ELO.

> **Information**
>
> In all cases, it is recommended that you always and exclusively transfer the SAP currency key for postings in SAP. A significant issue related to exclusively using ISO codes is the fact that the same ISO code can technically be assigned to multiple SAP currencies. It is also possible that no ISO code is assigned to an SAP currency.

'PurchaseOrderItems' mapping

The *PurchaseOrderItems* mapping is used to subsequently load purchase order items into the invoice form. This mapping is called in the background by a form script, which then assigns the results list to the item lines. Besides the relevant master data fields, it is possible to influence the output of the linked SAP search help */ELO/SH_PO_ITEMS* using specific input parameters:

- EXCL_STAPO (*flag*): If enabled, the results list does not contain any statistical order references (see table *EKPO*, field *STAPO*).
- CALC_QTY (*flag*): If enabled, only the open delivery quantities per item are returned. Open delivery quantities are quantities already received but for which an invoice has not yet been posted. If disabled, the total quantity of the purchase order is returned.
- EXCL_USED (*flag*): If enabled, purchase order items that have already been fully invoiced are excluded from the selection. "Fully invoiced" means that incoming invoice(s) have been

posted for the total quantity of goods received. Consequently, it only makes sense to use it in conjunction with enabled *CALC_QTY*.

The search help also returns the field *DISTRIB*, which contains the account assignment category of the purchase order item, see Multiple account assignment of purchase order items (MM).

The mapping for individual selection of purchase order items via the dynamic keyword list in the item table also accesses the SAP search help */ELO/SH_PO_ITEMS*. The explanations in this subchapter apply equally in this case.

Mappings for partner bank types

Partner bank types are linked to various SAP search helps per account type via the mappings *PartnerBankVendor* (vendors) and *PartnerBankCustomer* (customers).

Depending on the SAP system (especially *S/4* systems), you may have to remove the filter on the search help field *TABNAME* in each mapping.

Mappings for customer and vendor accounts

The mappings *Vendor* and *Customer* are provided for vendor and customer numbers. The SAP search helps provided specifically for this purpose offer an extended field selection compared to the respective standard search helps.

In the standard software version, accounts are filtered according to the CpD indicator (Invoice data form):

- If the CpD indicator is set, only CpD accounts are displayed.
- If the CpD indicator is not set, only accounts are displayed that are not CpD accounts.

Integrated SAP search helps

As you have already learned in earlier subchapters, the SAP transport for Template Invoice includes some SAP search helps in the namespace */ELO/*. These allow you to extend the functionality of standard search helps. In terms of configuration and use, these search helps behave in the same way as elementary search helps in standard SAP. You will find all integrated search helps either in SAP directly or in *searchHelps.config* under the respective mapping. These are recognizable in both cases by the aforementioned namespace prefix of the SAP search help.

Localized date formats

When you call search help mappings (unlike function module mappings), date values are transferred from the form in a localized format based on the user (e.g. *07/25/2020* [US] vs. *25.07.2020* [EN]). To ensure that the search help call is executed in a date format that SAP can understand, the header script extension of Template Invoice writes the respective date format to the hidden map field *LCL_DATE_FORMAT*. This field is used by all search help mappings for date interpretation and can also be used for custom mappings.

Example: To filter cost centers by validity periods, the *CostCenter* mapping transfers a from and to date as filter criteria. A Handlebars expression converts the local booking date into a date that is compatible with SAP.

```
{{{formatDate "YYYYMMDD" IX_GRP_ERP_BOOKING_DATE IX_MAP_LCL_DATE_FORMAT}}}
```

Surcharge search helps

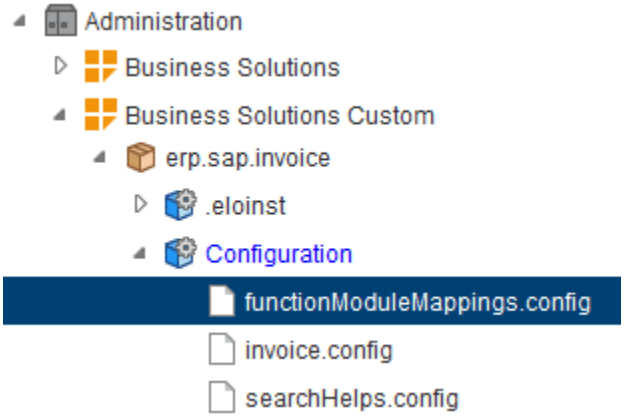
The surcharge table in the Invoice form contains fields that have the same description as in the item table but technically have different variable names. This means that all fields in the surcharge table must use their own search mapping. In terms of content, these search helps usually do not differ from their counterparts in the item table. For example, the *Cost center* field calls the same SAP search help with the same filter criteria in both cases. Only the focus fields, and possibly other fields to which data is written back or which are used as filters, differ in the mappings. Accordingly, when you modify a search help mapping, you also need to consider the respective counterpart in the item or surcharge table. The search help mappings for the surcharge table have the suffix *ForCharges*, e.g. *TaxCodesForCharges*.

Search helps in different form modules

ELO Invoice contains a variety of form modules, such as different role-based views of the invoice document.

Not all form modules that contain fields with a reference to SAP are linked to search helps in Template Invoice (e.g. module *611_position_data_compact*). If required, SAP search helps can be linked to customer search helps in the implementation project. In most cases, you can use the fields of the item table (module *619_position_data_ext_sap*) as a template.

Function module mapping



This section requires you to have basic knowledge of the function module mapping and its configuration as well as the functionality of SAP function modules.

Function module mappings in Template Invoice serve the following purposes:

-

To read additional data from SAP relevant for processing incoming invoices (e.g. order items), especially during background processing (e.g. in the workflow) and generally for complex data structures.

- To generate or simulate transaction data in SAP. Specifically, Template Invoice primarily uses function modules to post or park different types of incoming invoices in SAP.

The following subchapters list all the function module mappings provided in the standard software package. Unique aspects are explained in detail for each mapping.

'docOpenItemsCheck' mapping

This mapping is for document tracking and is used by the ELOas rule *DocumentState*. It addresses the SAP function module */ELO/GET_INVOICE_STATE*.

The function module receives the document number of a parked FI document or an MM document and returns, if available, the number and year of the corresponding follow-up FI document used to create an open item on the respective vendor or customer.

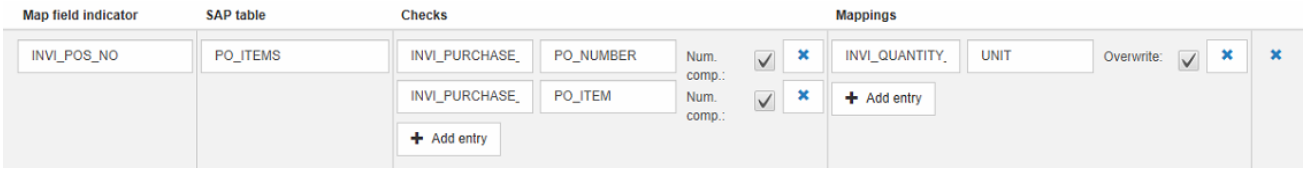
'docBalancedCheck' mapping

The *docBalancedCheck* mapping works in the same way as *docOpenItemsCheck*, except that it searches the clearing document for the open item. If it is found, the document number and year are written back separately. For details, see 'docOpenItemsCheck' mapping.

'getItemData' mapping

This mapping downloads purchase order items from SAP. In the standard version, this mapping is called in the subworkflow *erp.sap.invoice.Download* after automatic analysis, usually by ELO DocXtractor. This calls the SAP function module *BAPI_PO_GETITEMS*.

If there is more than one purchase order for the incoming invoice, we end up with a unique parameterization: The BAPI does not offer multiple selection via the purchase order number. To prevent the mapping being called repeatedly, all purchase orders for the entered vendor are loaded instead. The relevant purchase order items are then filtered out with the help of the *advanced data transfer* function.



Since the order unit is also required, the field is mapped under *Mappings*. You can also map other fields from this table here if required.

> **Please note**
>
> Return tables that are filtered using the advanced data transfer function should normally not have field mappings for the respective interface parameter, since the filter criteria are not

applied here. Instead, all relevant interface table fields should be linked to the target field in ELO in the *Mappings* area.

'invoicePost' mapping

MM invoices are posted directly with the mapping *invoicePost*, which calls the function module *BAPI_INCOMINGINVOICE_CREATE*. Since this BAPI has a similar function to the SAP transaction *MIRO*, it can be used for a variety of business processes in the context of processing MM invoices. Accordingly, there is a wide range of parameterization options that must always be evaluated on an individual customer basis. For details on the functionality and parameterization of *BAPI_INCOMINGINVOICE_CREATE*, refer to the corresponding SAP documentation or SAP Notes.

The basic structure of the relevant BAPI interface parameters in the standard software package is as follows:

IMPORTING

- HEADERDATA: Document header data, e.g. booking date, gross amount, invoicing party
- ADDRESSDATA: Address of the invoicing party and identification of one-time accounts (CpD)

TABLES

- ITEMDATA: Item data without account assignment, e.g. quantities (units), amounts
- ACCOUNTINGDATA: Account assignment data per item, complementary to *ITEMDATA*
- GLACCOUNTDATA: Item data without reference to a purchase order item, used in Template Invoice for surcharges

Success or error messages can be retrieved using the BAPI table *RETURN*. After successful posting, the document number (*INVOICEDOCNUMBER*) and the fiscal year (*FISCALYEAR*) of the created posting are returned.

'invoicePark' mapping

This mapping is used to park an MM invoice and calls *BAPI_INCOMINGINVOICE_PARK*. The parameterization is identical to *BAPI_INCOMINGINVOICE_CREATE*, see *invoicePost* mapping.

You can then later post the parked document manually in SAP or automatically via *BAPI_INCOMINGINVOICE_POST*. This process is not part of Template Invoice.

'invoiceSave' mapping

This mapping is used to park an MM invoice and calls *BAPI_INCOMINGINVOICE_SAVE*. It generates the body of an MM invoice, i.e. when this mapping is called, an invoice without item data is parked. This is especially useful or necessary in more complex scenarios if the item lines cannot be (easily) determined in ELO and an SAP standard transaction (e.g. *MIRO*) needs to be used instead.

This is the case, for example, with order items with multiple account assignments, i.e. purchase order items that allocate the bookable quantities to different account assignment objects according to specific rules. When the purchase order items are loaded with the function module mapping

getItemData or the search help mapping *PurchaseOrderItems*, the distribution type (SAP field *DISTRIB*) is written back for each item. If this is the case, the invoice posting in the standard workflow is rejected with this mapping (*invoiceSave*). To ensure backwards compatibility, you need to enable this behavior once, see Multiple account assignment of purchase order items (MM).

When transferring the invoice body, it is irrelevant whether the invoice has other items with single account assignment. As soon as the invoice has an item with multiple account assignment, no item data is transferred, not even of items with single account assignment. However, the actual purchase orders and associated items are transferred as suggestions. When you open the generated document in SAP (e.g. via *MIRO*), it is therefore possible to determine the suggested purchase order items automatically. The rules for multiple account assignment per item are also applied. The invoice can now be posted in SAP.

If, however, a customer project requires complex derivations (such as the case described above) to be made directly in ELO, this requirement must be evaluated and implemented in the customer project. Template Invoice does not include the application of SAP distribution rules for purchase order items with multiple account assignments in ELO.

'invoicePostFI' mapping

FI invoices are posted directly with the mapping *invoicePostFI*, which calls the function module *BAPI_ACC_DOCUMENT_POST*. Since this BAPI has a similar function to the SAP transaction *FB01*, it can be used for a variety of business processes in the context of processing FI invoices. Accordingly, there is a wide range of parameterization options that must always be evaluated on an individual customer basis. For details on the functionality and parameterization of *BAPI_INCOMINGINVOICE_CREATE*, refer to the corresponding SAP documentation or SAP Notes.

The basic structure of the relevant BAPI interface parameters in the standard software package is as follows:

IMPORTING

- DOCUMENTHEADER: Document header data, e.g. booking date, vendor, company code
- CUSTOMERCPD: Additional data (especially address) for the one-time account (CpD)

TABLES

- ACCOUNTGL: Item data for GL account lines
- ACCOUNTRECEIVABLE: Item data for customer lines
- ACCOUNTPAYABLE: Item data for vendor lines
- ACCOUNTTAX: Tax lines
- CURRENCYAMOUNT: Amounts and currencies for the above item lines
- EXTENSION2: Additional data for *BADI_ACC_DOCUMENT*, see also chapter BAdIs

This mapping uses the posting line script as a preprocessor script (*RF_erp_sap_invoice_service_GetPostingLines*, available in expert mode under *Preprocessing IX function*). The purpose of the posting line script is to separate the form information, usually only available in header and item data, into the different table structures of the BAPI (see above). In addition, the posting row script makes some case distinctions, such as setting suitable parameters

for BAdI processing (*EXTENSION2*). Use of Handlebars expressions alone in MAPLOOPs is not workable here.

With table-type structures, the output object of the posting row script is accessed in the mapping process using Handlebars in *OBJ_PROP_LOOP*.

> **Example**
>
> GL account lines prepared by the posting row script are stored as objects below *preObj.accountgl*, each with a consecutive line key.
>
> The BAPI table *ACCOUNTGL* is populated by *OBJ_PROP_LOOP* via *preObj.accountgl*. For each child object (line), the individual fields are accessed via Handlebars, with `{I}` acting as a placeholder for the current line in the loop.
>
> ```
> {{{preObj.accountgl.{I}.gl_account}}}
> ```
>
> accesses the GL account of the current line, for example.
>
> This example also illustrates that the actual JSON key (property) of each child object (line) has no further relevance when mapping in the *OBJ_PROP_LOOP*, since it is represented by the placeholder. Moreover, the key is a technical necessity for the preprocessor script.

The *EXTENSION2* table is used to transfer control and user data to the provided BAdI. If you don't want this (e.g. due to conflicts with other BAdI implementations which cannot be resolved using suitable filter criteria), you must disable the transfer of the table entirely. Doing so means that the functionality described in chapter BAdIs is no longer available.

'invoiceParkFI' mapping

This mapping is used to park an FI invoice and is largely identical to the mapping for direct posting, see *invoicePost* mapping. It addresses the same function module with almost identical parameterization. The main difference occurs in the field *DOCUMENTHEADER - BUS_ACT* (business transaction), which controls parking (instead of direct posting).

You can then later post the parked document manually in SAP. A BAPI for posting parked FI documents is not provided, see SAP Note *2092366*. If necessary, you should check other automation options (e.g. batch input, [custom] function module) or the initial direct posting from ELO in the customer project.

The *EXTENSION2* table is used to transfer control and user data to the provided BAdI. If you don't want this (e.g. due to conflicts with other BAdI implementations which cannot be resolved using suitable filter criteria), you must disable the transfer of the table entirely. Doing so means that the functionality described in chapter BAdIs is no longer available.

> **Information**

The SAP-side requirements for using *invoiceParkFI* are described in SAP Note *2092366* and its dependencies. If necessary, implement this note and the dependent notes.

'invoiceCheckFI' mapping

This mapping is used to simulate an FI invoice and is largely identical to the mapping for direct posting, see *invoicePost* mapping. The function module called here is *BAPI_ACC_DOCUMENT_CHECK* and is used to validate the respective parameters of an FI posting. The function module does not generate an FI document but only simulates a posting. Error or success messages are found in the table *RETURN*, as with *BAPI_ACC_DOCUMENT_POST*.

The *invoiceCheckFI* mapping is not used in the standard version and can be integrated at an appropriate point if required, such as in a custom workflow. A subworkflow for the posting simulation is also provided: *erp.sap.invoice.Simulation*.

Please note

In most cases, the *invoiceCheckFI* call is followed by a call from a posting mapping (e.g. *invoiceParkFI*). Therefore, when you change a posting mapping, you need to check whether the respective change must also be made for *invoiceCheckFI*, and vice versa. Otherwise, the check result could differ from the posting result because the parameterization is different.

The *EXTENSION2* table is used to transfer control and user data to the provided BAdI. If you don't want this (e.g. due to conflicts with other BAdI implementations which cannot be resolved using suitable filter criteria), you must disable the transfer of the table entirely. Doing so means that the functionality described in chapter BAdIs is no longer available.

'invoiceTaxCalculate' mapping

The *invoiceTaxCalculate* mapping is used in SAP to calculate tax on FI documents. The BAPIs for posting/parking FI invoices do not offer automatic calculation of tax document item lines but instead expect them in complete form (see parameter table *ACCOUNTTAX*).

In simple cases (e.g. VAT on national transactions), the tax item lines can be generated directly in ELO. Necessary additional data, such as the tax account or the transaction key, is provided by the search help mapping *TaxCodes*.

With complex tax transactions, such as those within the EU, you need to calculate the tax in SAP. For this purpose, SAP provides the *FI_TAX_SERVICES_CALCULATE* function module as an addition to the BAPIs for FI postings. The interface parameters are the same as for *BAPI_ACC_DOCUMENT_POST* (e.g. *invoicePostFI* mapping). All *CHANGING* parameters can be changed by the function module. In particular, the generated tax lines are returned in the table *ACCOUNTTAX* and their amounts in the table *CURRENCYAMOUNT*. If an error occurs, the function module does not make any changes to the document but simply returns corresponding messages in the *RETURN* table along with an error code (*RETURN_CODE*).

Information

The SAP-side requirements for using *invoiceTaxCalculate* are described in SAP Note *2712238* and its dependencies. If necessary, implement this note and the dependent notes.

Technical background:

When selected in the form, the search help mapping *TaxCodes* automatically identifies complex or multi-level tax codes according to a flag in the search help output. As soon as a tax code in the document has this flag, all tax lines (including those for simple tax transactions) must be calculated in SAP. In this case, the posting line script (which is executed when *invoicePostFI* is called) also executes the mapping *invoiceTaxCalculate*. The tax lines returned by SAP are transferred to the posting line script output and then transferred to the respective BAPI along with the remaining document data. If errors occur when calling *FI_TAX_SERVICES_CALCULATE*, these are forwarded to the calling mapping and updated accordingly on the SORD. The document is not transferred to SAP.

Information

Error handling when calling *invoiceTaxCalculate* within the posting line script has been carried out since version 1.00.010 of Template Invoice. In older versions, the program attempts to post or park the document and all errors are discarded.

The *EXTENSION2* table is used to transfer control and user data to the provided BAdI. If you don't want this (e.g. due to conflicts with other BAdI implementations which cannot be resolved using suitable filter criteria), you must disable the transfer of the table entirely. Doing so means that the functionality described in chapter BAdIs is no longer available.

Status fields for transfer

The respective SORD has the following additional map fields for controlling the workflow:

- ERP_SAP_INVOICE_TRANSFERRED: Is true (field value *1*) if transfer to SAP was successful
- ERP_SAP_INVOICE_TRANSFERRED_TYPE: Specifies whether the document was parked (*PARKED*) or posted directly (*POSTED*)

The purpose of the fields is, among other things, to prevent the same document from being transmitted/posted multiple times.

Advanced data transfer via Handlebars syntax

Field type	Field name	Handlebars string
Index group field	INVOICE_NO_ERP	{{{result.E.INVOICEDOCNUMBER}}}{{{result.E.FISCALYEAR}}}
Map field	ERP_SAP_INVOICE_TRANSFERRED	{{#if result.E.INVOICEDOCNUMBER}}1{{/if}}
Map field	ERP_SAP_INVOICE_TRANSFERRED	{{#if result.E.INVOICEDOCNUMER}}POSTED{{/if}}

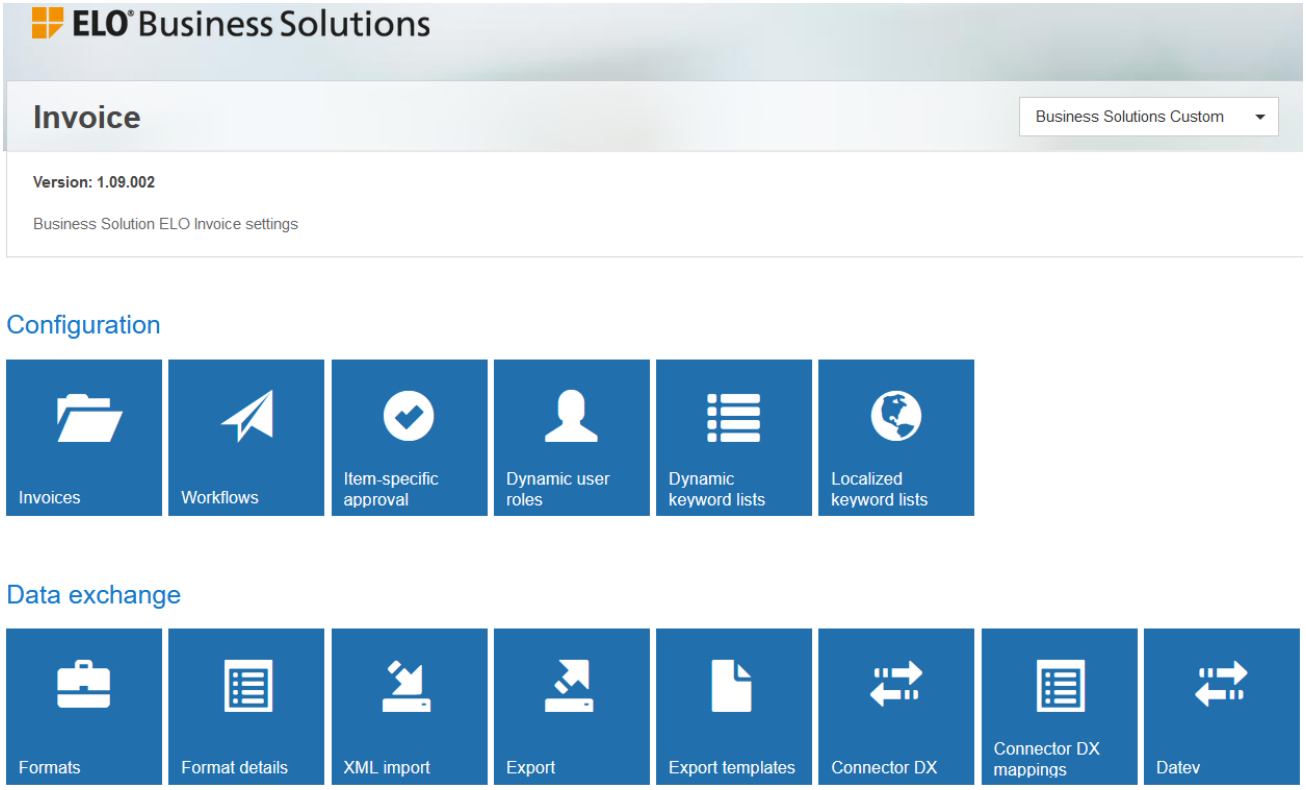
In principle, these fields are solely managed, written, and read by the workflows provided. As an additional security feature, all posting or parking function module mappings also write directly to these fields.

In the event of a malfunction in the workflow controller (e.g. due to server hardware overload), this reduces the probability of unintentional multiple transfers.

ELO DocXtractor customizations

Navigate to Invoice management in the ELO Administration Console.

You will see the *Connector DX mappings* tile.



Since Template Invoice adds the *Credit note* check box to the form and the *Invoice correction* check box exists as the equivalent in DocXtractor Verifier, matching between DocXtractor and ELO must be done via the *connector_dx* package. The following extension transfers the check mark from the Verifier to the ELO form.

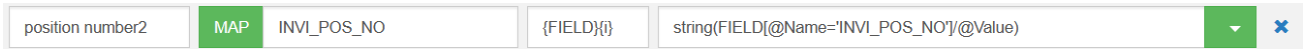
In addition, DocXtractor recognizes whether the invoice has a purchase order reference or not. If DocXtractor finds a purchase order, this information can be transferred from the Verifier to ELO.

Complete the entries in the *Mappings* tab.

with po reference WITH*PO*REFERENCE string(/DXDOC/FIELD[@Name='INV*IS*MM']/@Value)

is credit note SAP*IS*CREDIT_NOTE string(/DXDOC/FIELD[@Name='INV*CREDIT*NOTE']/@Value)

Based on the transferred information, the SAP data associated with the purchase order is subsequently read and transferred to MAP fields. In the *Table mappings* tab, add a new mapping for the *line items* entry.



position number2 INVI*POS*NO {Field}{i} string(FIELD[@Name='INVI*POS*NO']/@Value)

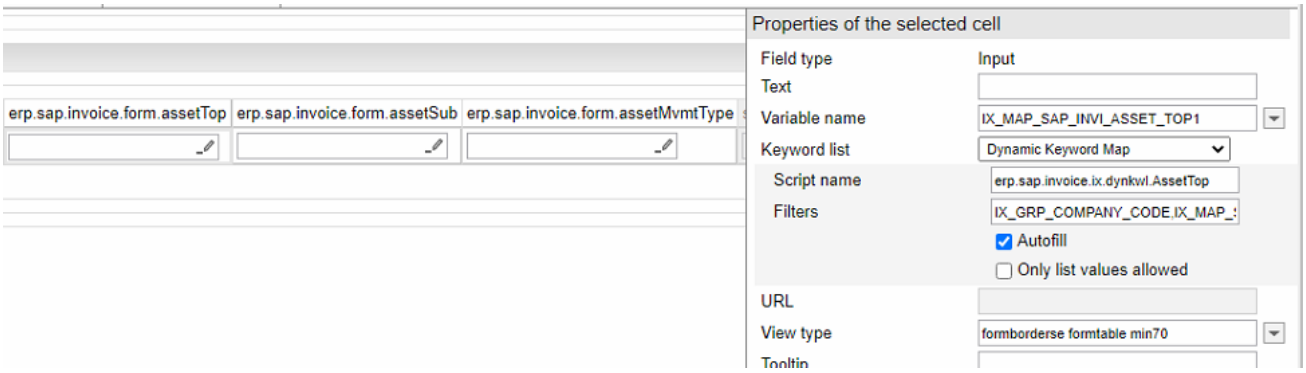
Special business transactions

Asset posting (FI)

The *Asset posting* business transaction describes how to assign a line item to an asset, specifying the asset number, subnumber, and transaction type. This business transaction differs from posting to a GL account with the additional specification of an asset; there is no assignment to a GL account. To process this type of posting, you need to import and enable the corresponding BAdI (see chapter BAdIs). This is because the parameter for the asset transaction type in the BAPI is missing, which means that an asset reconciliation account is not derived. The BAdI sets the transaction type and derives the reconciliation account. It gets the transaction type from the enhancement structure *EXTENSION2*, which is entered in the posting row script.

Template Invoice contains several components that enable you to easily activate asset posting. Generally, all you need to do is show the additional fields.

Show additional fields



In the *sol_invoice* form, the following columns of the item table must be displayed under module *619_position_data_ext_sap*:

- erp.sap.invoice.form.assetTop: Asset number
- erp.sap.invoice.form.assetSub: Asset subnumber
- erp.sap.invoice.form.assetMvmtType: Asset movement type

Asset depreciation area

Part of the derivation of the asset reconciliation account described in the following subchapter is the restriction of a posting to the relevant depreciation area. The default value is depreciation area *01*. If a customer system uses a different depreciation area, it has to be transferred to the BAdI as an additional parameter via *EXTENSION2*. We recommend that you customize the posting line script for this purpose. In the script, you can address the function *setDeprArea*.

Use and functionality

When assigning to assets, the three additional fields in the respective item line must be completed. All fields are already linked to the corresponding SAP search helps in the standard version.

If the asset transaction type has been entered in item lines in the form, the posting line script interprets these as an asset posting. The transaction type is set with reference to the original item as a table line of *EXTENSION2*. You can see the exact parameterization in the *setAssetMvmtType* function of the posting line script. During BAdI processing, the following steps are carried out for each line item with an asset transaction type:

- If no GL account is set, the asset reconciliation account is derived:
- Derivation of the *account determination* based on company code, asset, and subnumber (table *ANLA*)
- Derivation of the *chart of accounts* based on the company code (table *T001*)
- Derivation of the *reconciliation account* based on the chart of accounts, account determination, and depreciation area (table *T095*)

> **Information**
>
> If a GL account was entered in the form, you post directly to this GL account.

- The transaction type set by the extension table is used.

Asset numbers and subnumbers are part of the BAPI table *ACCOUNTGL* and are also assigned there. Therefore, these fields do not need to be transferred via *EXTENSION2*.

Complex tax transactions (FI)

Unlike simple tax transactions, there are some tax codes that require you to determine multiple tax rates in document item lines. For example, invoices within the EU are usually posted with the same amount of output and input tax with a zero balance. In this example, one tax code therefore requires two tax item lines.

Template Invoice automatically recognizes multi-level tax transactions and allows SAP to calculate them before posting. You can recognize these tax codes by the indicator *Calculate tax in SAP* in the dynamic keyword list for tax codes and after selection by the field with the same name in the respective item line. As soon as the item lines contain at least one such tax code, the field with the same name is also enabled in the footer data.

In these cases, the tax percentage rate in the forms is always 0.00%, since the effective tax rate cannot be calculated in ELO. However, the effective tax rate is often 0.00% anyway (as in the example above) since the taxes are entered as a zero balance.

Technical details and relationships are described in chapter 'invoiceTaxCalculate' mapping.

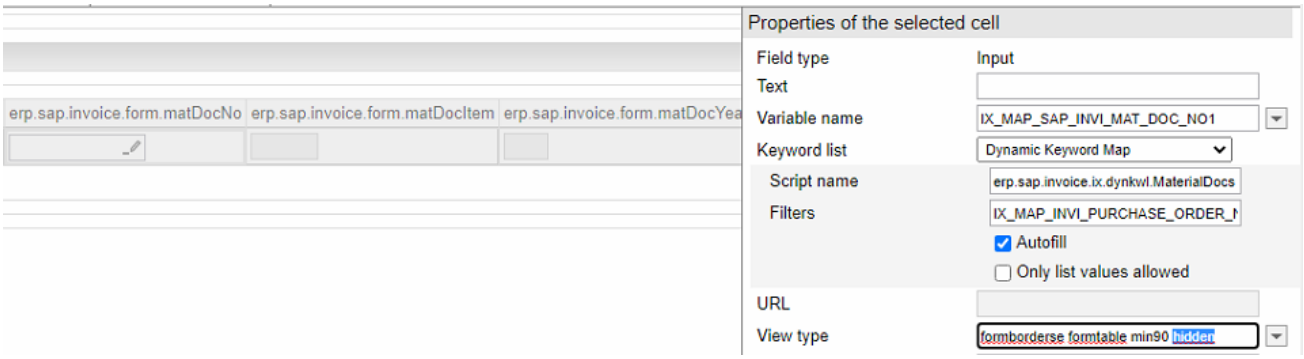
> **Information**

If you always want the tax to be calculated in SAP (even with simple tax transactions), the corresponding field in the footer data must always be enabled. It is not enabled by default in Template Invoice and is not recommended as it would cause all tax information in ELO to be lost.

Assignment of goods receipt (MM)

There may be cases where you want or need to assign multiple goods receipt documents to an MM invoice item. The recommended solution is to use ELO DocXtractor for (automatic) assignment based on advanced purchase order history data.

For simple cases involving direct mapping of the invoice item and goods receipt document, the item table in the form provides hidden fields for material document, document item, and fiscal year.



In the *sol_invoice* form, the following columns must be displayed under module *619_position_data_ext_sap*:

- erp.sap.invoice.form.matDocNo: Material document number (goods receipt)
- erp.sap.invoice.form.matDocItem: Item number in the material document
- erp.sap.invoice.form.matDocYear: Fiscal year of the material document

The field for the material document number is linked to a search help and automatically completes the other two fields according to the selected material document item. Mapping is done in the *ITEMDATA* table of the *invoicePost* and *invoicePark* function module mappings.

Multiple account assignment of purchase order items (MM)

Background

Template Invoice does not provide a function for automatically allocating quantities in the case of purchase order items with multiple account assignments. The posting process is instead controlled by a workflow using the *invoiceSave* function module mapping. For more details, see chapter 'invoiceSave' mapping.

Functionality

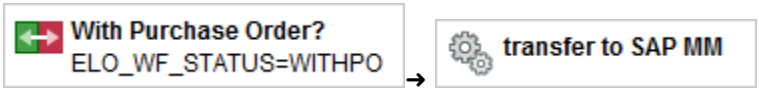
The transfer produces a parked invoice body without invoice items, but with a selection of the purchase order items assigned in the form. After you open the parked invoice in SAP (e.g. via transaction code *MIRO*), you can load the purchase order items and quantities and allocate them automatically according to the account assignment rules for each item. Afterwards, you can post the invoice in SAP.

Accordingly, you do not need to enter item data in the form. If you do enter information (e.g. invoiced quantities per item), it is lost during transfer and has to be entered again in SAP.

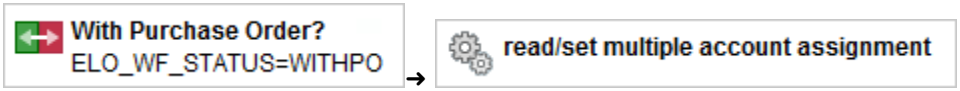
Activation

The invoice body is ready for transfer in Template Invoice but is not yet enabled. All you need to do is to modify the workflow *erp.sap.invoice.Transfer*:

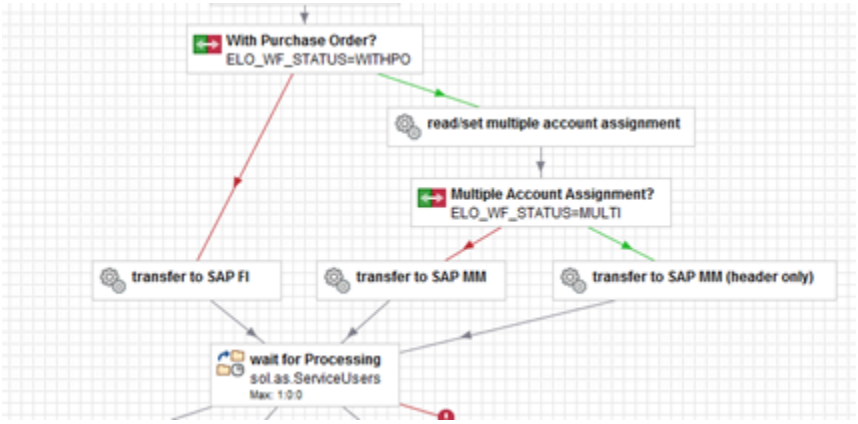
1. Remove the connection:



2. Add the connection:



Afterwards, the relevant part of the workflow should look like this:



Direct tax (FI)

This business transaction applies to invoices that include further allocation of taxes incurred. This may be the case, for example, if insurance companies charge the insured party for the tax portion of a repair service rendered by a third party. From an accounting point of view, this type of invoice line item is a direct tax entry, i.e. a direct posting to a tax account without reference to an item with a tax base amount (this is only included on the original third-party invoice).

Tax accounts can usually be posted directly in SAP, provided that a base amount corresponding to the respective tax rate is specified. Posting with *BAPI_ACC_DOCUMENT_POST* works in the same way except that with this method, the direct tax lines must be explicitly marked. Unlike standard GL account lines, direct tax lines must therefore be stored in the BAPI table *ACCOUNTTAX* and assigned the code *DIRECT_TAX* and a *tax base amount*.

The tax category of the selected GL account is written to a hidden map field per item line using the search help mapping *AccountGl*. If the tax category corresponds to the < or > (lower or higher than) characters, the posting line script recognizes the account as a tax account and the line item is recognized as a direct tax line. In the unlikely event that the customer system requires a different set of tax categories to identify a tax account, the posting line script needs to be modified accordingly.

The tax base amount is automatically determined based on the percentage of the tax code and the respective item line is redirected to the BAPI table *ACCOUNTTAX*.

If you do not want this to happen automatically, you need to prevent the tax category being written back to the field *IX_MAP_INVI_GLA_TAX_CAT* in the aforementioned search help mapping or you need to redirect the item to another field.

Alternatively, you can index direct tax items per line with the map field *INVI_SAP_DIRECT_TAX*. The corresponding column of the item table can be displayed for the purpose of manual entry (form *sol_invoice*, module *619_position_data_ext_sap*, column *erp.sap.invoice.form.directTax*). If *INVI_SAP_DIRECT_TAX* in an item line is *true*, the posting line script also treats this line like a direct tax line (see above). You can also use a tax category and *INVI_SAP_DIRECT_TAX* at the same time. Only one of the conditions has to be met to trigger a direct tax posting.

Cash discount net procedure (FI)

If you specify net document types, the net procedure is not automatically applied with the *BAPI_ACC_DOCUMENT_POST* call. The provided BAdI enables you to apply the net procedure automatically. Go to the *Accounting* tab of *invoice.config* and enable the option *Cash discount: Automatic net procedure* (see chapter invoice.config – Accounting). Once you have done this, the BAdI is given a corresponding flag via the *EXTENSION2* table (for technical details, see the *enableAutomaticNetProcedure* function in the posting line script).

If the BAdI finds a net document type and the automatic net procedure is enabled, the document is posted using the net procedure. The functionality of the SAP transaction *FB01* for net document types serves as a model. We cannot guarantee that the FB01 logic is fully congruent with every customer system at this point. The posting result using the net procedure needs to be validated in every case in the respective SAP system. You may be required to change the BAdI logic, e.g. by copying the BAdI provided.

Network posting

The item table contains fields that allow you to do postings to network and network activity.

In the *sol_invoice* form, the following columns must be displayed under module *619_position_data_ext_sap*:

Column heading	Map field	Meaning
erp.sap.invoice.form.networkTop	SAP_INVI_NETWORK_TOP	Network plan
erp.sap.invoice.form.networkSub	SAP_INVI_NETWORK_SUB	Network plan transaction

There is already a search help linked to a dynamic keyword list. The fields are already assigned in the relevant function module mappings provided in the standard package.

FBS1: Accrual/deferral posting with reversal date (FI)

This business transaction applies to the accrual/deferral posting of an FI invoice, specifying a reversal date and a reversal reason, see SAP transaction *FBS1*. The purpose of this type of posting is to automatically reverse the document in SAP on a key date (often at the end of a month or year).

BAPI_ACC_DOCUMENT_POST does not provide a function for setting the reversal date and the reversal reason. The BAdI provided offers the option to transfer both fields using the BAPI table *EXTENSION2*. If a document has the reversal date *31.12.2020* and *04* as the reversal reason, this results in the following parameterization of *EXTENSION2*:

	STRUCTURE	VALUEPART1	VALUEPART2	VALUEPART3	VALUEPART4
(Line 1)	/ELO/EXT_ACC_DOC	REV_REASON		04	
(Line 2)	/ELO/EXT_ACC_DOC	REV_DATE		20201231	

The posting line script has two functions for this purpose: *setRevReason* (reversal reason) and *setRevDate* (reversal date). These functions are not called by default in Template Invoice and must be implemented in the customer project if required.

Posting to accounts receivable (FI)

In the standard version of Template Invoice, you can specify a vendor but not a customer. Template Invoices provides components for posting incoming invoices to customer accounts as debit or credit items. It is not possible to post a conventional customer invoice, i.e. an A/R invoice to a customer account.

Form customization

In the *sol_invoice* form under the optional module *239_vendor_sap_with_customer*, you will find the additional field *Customer*. To switch the form modules, go to the tab group *004_accounting* in the form designer. Remove the assignment to *238_vendor_sap* and insert *239_vendor_sap_with_customer* in its place.

Use and functionality

The customer field is now displayed in the form under the vendor number and is linked to a search help that works in the same way as the search help for vendors. This also restricts the results, e.g. to one-time accounts (CpD) provided that the corresponding indicator is set. If there is an entry in the customer field, the vendor field should be empty (and vice versa). If both fields contain entries,

the vendor entry takes precedence and the customer entry is ignored. This process takes place in the posting line script. In the relevant function module mappings, the field mapping for customer lines is provided for in the *ACCOUNTRECEIVABLE* table.

Partner bank type

The field *Partner bank type* in the invoice data (module *239_vendor_sap_with_customer*) refers to the search help for vendor partner bank types. If you also want to be able to select customer partner bank types, you need to create an additional field *Partner bank customer* based on the original field. The search help mapping *PartnerBankCustomer* can be linked to the new field.

Noted items

Noted item postings have purely informative value. They are not displayed in your accounts but are only to remind you of outstanding payments due or to be made.

The posting does not update the account balance, and is merely managed as a line item in the open item account and the special G/L account.

You can view the noted items for the respective vendor with the transaction *FBL1N* if it is a credit-side posting.

If you have enabled the *Noted item* function, a noted item is created in the SAP system when the ELO workflow is started. This noted item remains in the SAP system until the ELO workflow has completed and the invoice posting has been closed. The noted item is subsequently cancelled.

To use the noted item function, you must have enabled the *Create noted item* option in *invoice.config* under *Accounts payable* in the *Accounting* tab.

> **Please note**
>
> This function requires customizing for noted items within the SAP system.
>
> The values listed here are only a guide and must be modified according to your customizing settings.

Field	Value	Meaning
Business transaction	RFST	FI statistical postings
Document status	S	Noted item K + P special G/L
ID	Z	Statistical vendor invoice
Posting key	39	Vendor credit posting

Information

Noted item attachments are linked to the *BKPF* SAP object within the SAP system. If you use the noted item, you must maintain a corresponding customizing setting in your SAP system for the *BKPF* object to match your document type.

Please note

Template Invoice is configured so that when clearing rejects a transaction, the noted item is cancelled in SAP. If it cannot be cancelled due to an error, the workflow is still terminated, and the noted item must be removed manually.

Alternatively, you can incorporate a corresponding logic into the template.

You will find the appropriate place in the *erp.sap.invoice.Base* workflow template.

Other

SAP payment locks: Data retention/search helps

The search help mapping *PaymentBlocks* accesses the SAP standard search help *DFKKLOCKS_LOCKR_4*, which sources data from the SAP table *TFK008*. In some SAP systems, this table may be empty or not up to date. In this case, SAP usually updates the payment locks in table *T008* instead. The search help then either returns no or unwanted values.

There are various solutions you can use:

- Change the search help mapping specified above to an SAP search help with access to table *T008* (e.g. *H_FARP_T008*)
- Migrate payment blocks from (tables) *T008*, *T008T* to *TFK008*, *TFK008T* using SAP report *FKKXPR04*.

> **Please note**
>
> This report performs the migration for all SAP mandates. Use this report at your own risk. Subsequent changes are not automatically synchronized so they must be repeated if necessary.

Required permissions

It is essential that the SAP interface users used in the context of Template Invoice have the necessary permissions specified in the Datatransfer documentation.

All other (especially technical) permissions for search help and function module calls vary depending on the scope of functions actually used and, if applicable, customer enhancements such as mapping additional search helps or function modules. Furthermore, additional permissions may be required after (minor) changes to the parameterization of a function module or search help. You should also note that the permissions required for the same operation may differ depending on the SAP system version or additional customer-side permissions checks may exist. For these reasons, this documentation does not contain a list of SAP permissions required for using Template Invoice.

If you want to define the permissions required for the interface user as precisely as possible, the *authorization trace* in SAP enables you to determine all required permission objects for a specific transaction (possibly in an iterative process). If in doubt, consult the person responsible for SAP or your SAP system specialist.