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

ELO Connectivity Pack for SAP® ERP – Datatransfer

Table of contents

ELO Connectivity Pack for SAP® ERP – Datatransfer

Getting started

The following provides you with information necessary before installing and using ELO Connectivity Pack – Datatransfer.

> **Information**
>
> This implementation has been tested for the following ELO versions:
>
> - ELOprofessional or ELOenterprise version 10.02.000 and higher
> - ELO Business Solutions Common 1.05.002 and higher

Security and licensing requirements

Depending on the configuration, you can use ELO Connectivity Pack – Datatransfer to transfer sensitive data from ELO to SAP. A secure permissions concept is therefore crucial. Importing data with RFC-enabled function modules can have implications for the SAP licenses associated with the connected system. If in doubt, consult SAP and/or the responsible IT systems specialist. ELO is not liable for any license violations.

Basics

The use of ELO Connectivity Pack – Datatransfer requires a corresponding SAP user with the necessary authorizations for executing RFC-enabled function modules (especially BAPIs).

> **Information**
>
> Definition of BAPI: A Business Application Programming Interface (BAPI) is a standard interface to the processes and data in the SAP system. BAPIs can be called within the SAP system from external application systems or other programs. BAPIs are the communication standard for business applications.
>
> Source: SAP help portal

An active and configured ELO Smart Link for SAP® ERP interface (referred to here as ELO Smart Link) between ELO and the SAP system is also required.

To use ELO Connectivity Pack – Datatransfer, your computer must also meet the following system requirements and release statuses of ELO and SAP:

- ELOprofessional/ELOenterprise version 10.02.000 and higher
-

ELO Business Solutions Common 1.05.002 and higher

- The following SAP versions or higher:
  - SAP_BASIS 702 SAPKB70213
  - SAP_BASIS 730 SAPKB73009
  - SAP_BASIS 731 SAPKB73106

Invoice SAP template

To better understand ELO Connectivity Pack – Datatransfer, a predefined template is available to all partners and customers. The *Invoice SAP* template was created in a separate package (*erp.sap.invoice*). This package contains the necessary settings for mapping an invoice receipt process, capturing and processing invoices in ELO, and subsequently transferring them to SAP. The variants for invoice receipt *with or without order reference* and the options *document parking and direct posting to SAP* are preconfigured as selectable scenarios. In addition to the function module mapping (*functionModuleMappings.config*), this package contains a large number of SAP search helps (*searchHelps.config*).

Installation

A license for ELO Connectivity Pack for SAP® ERP includes two functions: *Indexdownload* and *Datatransfer*. ELO Connectivity Pack – Datatransfer contains two packages:

- An SAP transport that has to be imported to the SAP target system.
- An eloinst file, which must be installed on the existing ELO system.

You will find the installation files on the ELO SupportWeb under:

Integration > SAP® ERP > ELO Suite for SAP ArchiveLink® > ELO Connectivity Pack for SAP® ERP > Downloads

SAP transport

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

The current SAP transport is available on the ELO SupportWeb at:

> **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 have been 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.

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

Common package

First, install *Business Solution Common 1.05.002* package or higher if it has not already been installed on your ELO system. This package provides the class framework, namespaces, and architectural concepts for services, actions, and function modules.

1. To start installation, log on to the ELO Java Client as Administrator. You can install the individual packages from the ELO SupportWeb with a single click.

Alternative: You can download the package containing the .eloinst files on the ELO SupportWeb. At this point, ELO Connectivity Pack – Datatransfer only requires the following two files:

- 00_sol.common_1.xx.xxx.eloinst
- Custom_sol.common_1.xx.xxx.eloinst

2. Next, install the current ELO Connectivity Pack for SAP® ERP – Datatransfer package: You will now find the configuration for ELO Connectivity Pack – Datatransfer in the *Business Solutions Custom* folder.
3. You can now proceed with the chapter ELO configuration.
   - erp.sap.conn_1.XX.XXX.eloinst
   - custom_erp.sap.conn_1.XX.XXX.eloinst

Installation information

When installing ELO Connectivity Pack – Datatransfer, the web app installation may not finish completely. You can tell that this is the case if there is no interface when you open the configuration. To fix this, switch 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.

In the ELO App Manager, the *App status* indicates whether the web app installation for Datatransfer 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.

ELO configuration

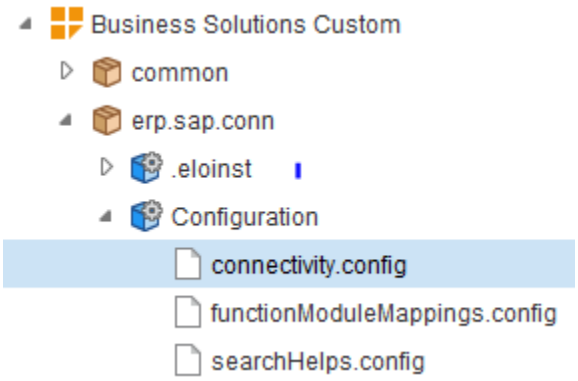

Fig.: ELO Business Solutions Custom; 'Configuration' folder

Once you have implemented the installation file with the *erp.sap.conn* package, you can go to *Configuration* > *connectivity.config*. The ELO Connectivity Pack – Datatransfer configuration interface opens in ELO.

Please note

Only edit the configuration in the *Business Solutions Custom* path to ensure the system can be updated in the future. When a module is updated, the configuration in the *Business Solutions* folder can be overwritten.

Information

You can carry out following configuration steps for ELO Connectivity Pack – Datatransfer under *Custom* in the *erp.sap.conn* package. For more comprehensive solutions, such as ELO Invoice, you can also create your own packages that contain the corresponding files. To get an interface, you need to create a ClientInfo in the ELO App Manager, such as for *erp.sap.invoice*.

ELO configuration – connectivity.config

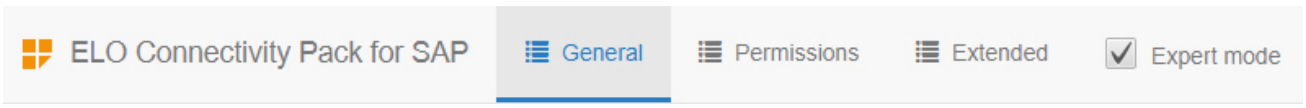

Fig.: Header in the configuration interface

The configuration interface consists of the areas *General* and *Permissions*. If you enable *Expert mode*, you will also see the *Advanced* area.

Information

When installing the solution, you may not be able to open the configuration interface. For more information, refer to the Installation information.

connectivity.config – General

ELO Connectivity Pack – Datatransfer uses the ELO Smart Link interface as the basis for communication with the SAP system. All ELO Smart Link instances that ELO Connectivity Pack – Datatransfer should have access to are configured under *General*.

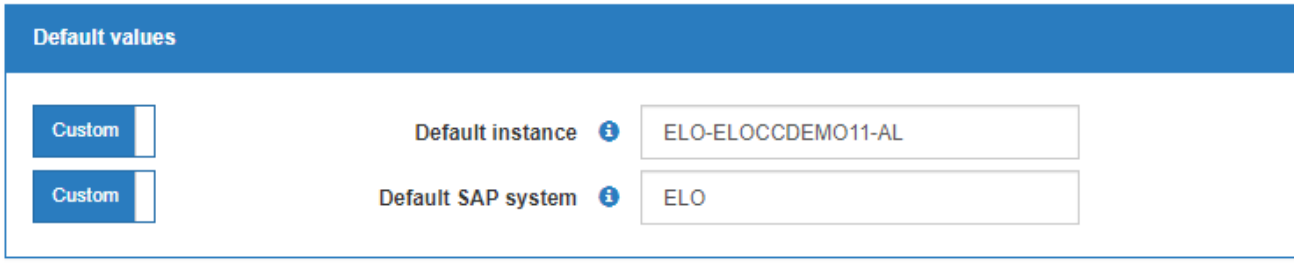

Fig.: Configuration interface, general

In the upper part of the page, you can enter default values that are used if you have not explicitly assigned an instance (e.g. *functionModuleMapping* or *searchHelps* see sections Configuration – searchHelps.config – search help or Configuration – Function module mapping).

Default instance: The name of a configured ELO Smart Link instance. This instance is used if no explicit instance is specified in the mappings.

Default SAP system: The system ID of the configured SAP system in the ELO Smart Link configuration. This must be located in the specified default instance. This instance is used if no explicit instance is specified in the mappings.

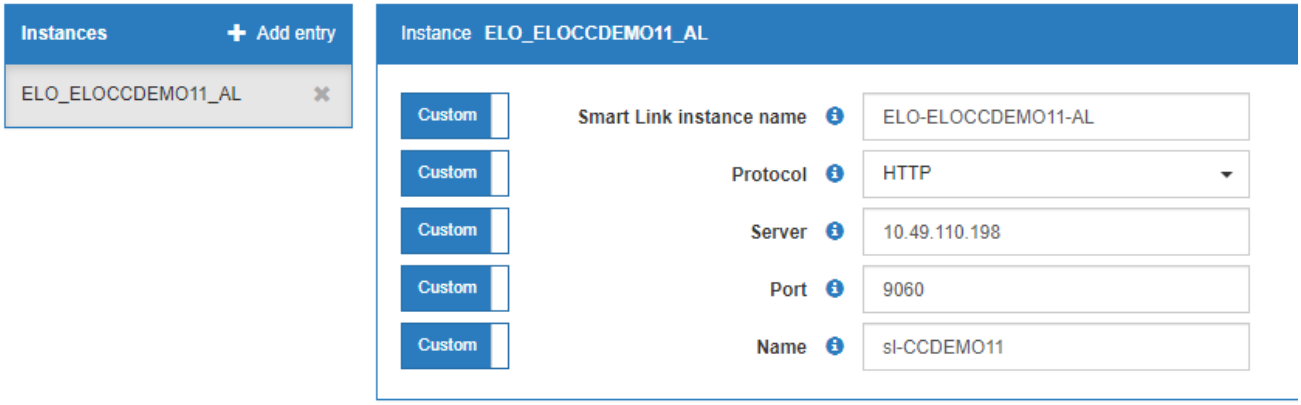

Fig.: Adding an instance

In the next step, you create and configure the ELO Smart Link for SAP® ERP instance.

Smart Link instance name: The name of a configured ELO Smart Link instance. The name must be identical to the name defined in the ELO Smart Link configuration.

Protocol: The protocol used for access. It must match the Tomcat configuration.

Server: Host name or IP address of the ELO Smart Link instance.

Port: The port of the ELO Smart Link instance is entered here.

Name: Name of the ELO Smart Link instance servlet.

The URL for access to the ELO Smart Link module contains the values entered above as components according to the pattern:

```
[Protocol]://[Server]:[Port]/[Name]
```

In the screenshot in the figure, the URL obtained is:

```
http://10.20.30.40:9060/sl_CCDEMO11
```

connectivity.config – Permissions

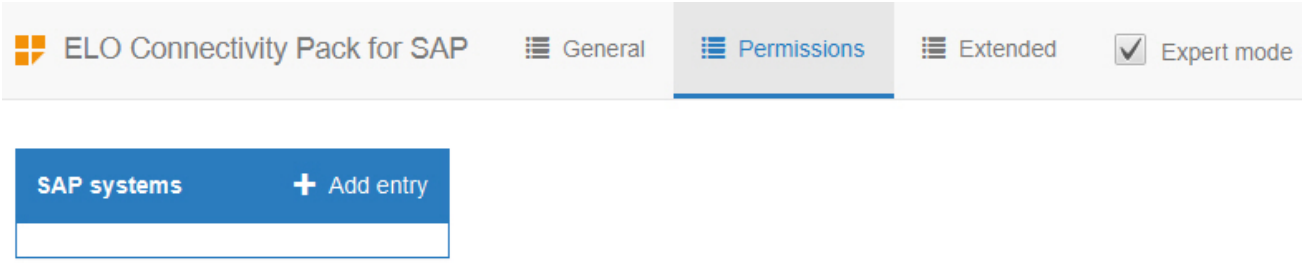

Fig.: Configuration interface, permissions

Under *Permissions*, you can restrict access to ELO Connectivity Pack – Datatransfer.

It is possible to restrict access to the SAP systems for certain user groups, for example.

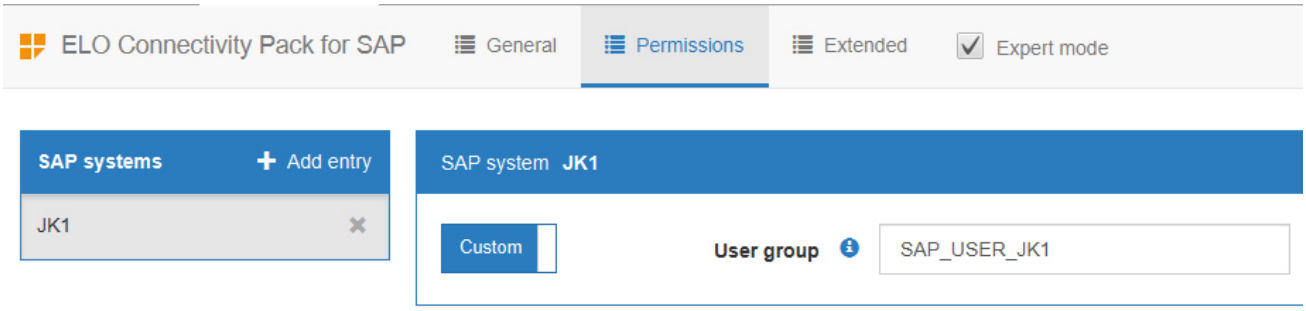

Fig.: Configuration interface, permissions assigned

1. To do this, make an entry whose key corresponds to the name of the SAP system linked to the ELO Smart Link instance.

2. Enter an ELO user group under the *User group* field.

Now, access to all function modules and SAP search helps associated with this system is restricted to the ELO user group entered.

connectivity.config – Expert mode – Advanced

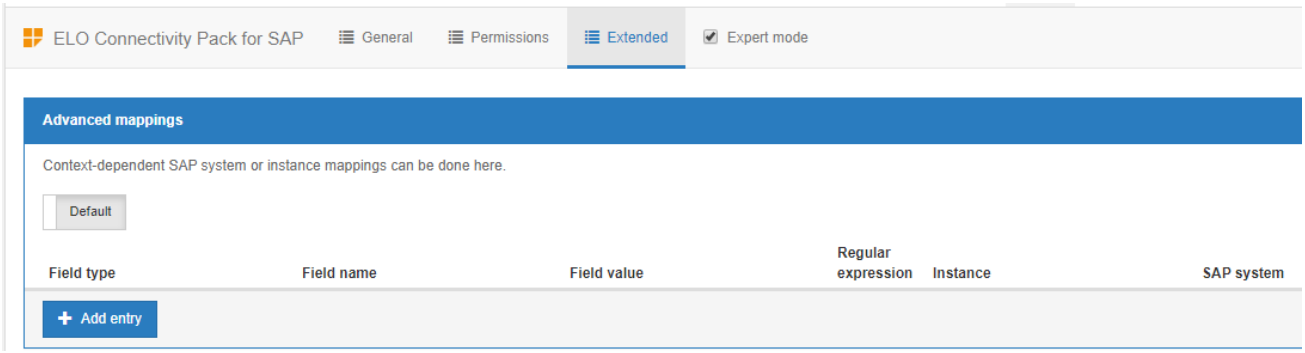

Fig.: Configuration interface, advanced

The settings under *Advanced* (Expert mode) are optional. In this area, you can define dynamic rules for SAP systems.

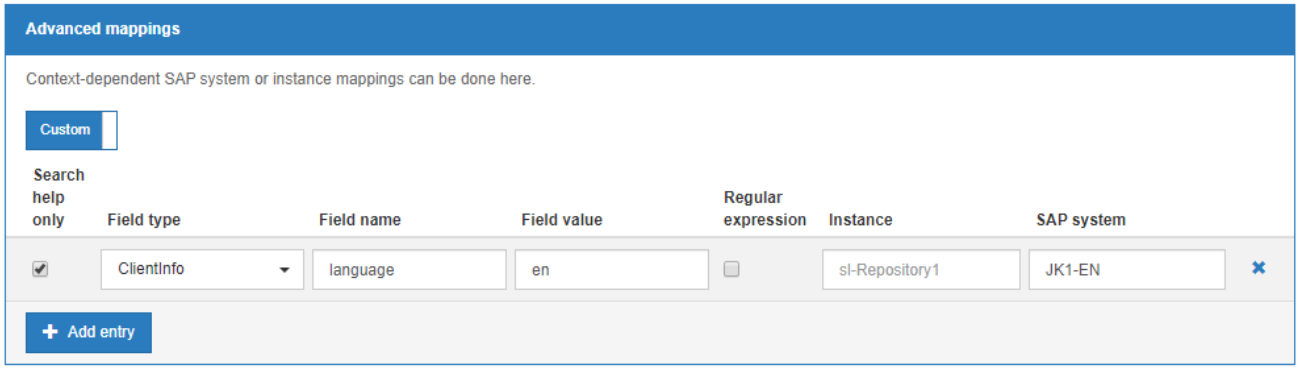

Fig.: Configuration interface, advanced

A possible use case is the connection of an SAP system in another language.

1. To do this create an *Advanced mapping*.
2. Select *Add entry* to create a new derivation rule.
3. You can then complete the following fields for the new entry:

   Search help only: If activated, this rule only applies to search help queries. Other queries (e.g. when executing function modules) are not affected by this rule.

   Field type: The field type determines the source of the information and, together with the field name (see below), makes up the information to be queried. The following values are possible:

   - ClientInfo: The ClientInfo object contains information about the user executing the respective application.
   -

GRP: With the field type *GRP*, the system requests an ELO metadata field.

Field name: The identifier of the field being requested, either from the set of metadata fields or from the ClientInfo object, depending on the field type. Example: The *language* field in the ClientInfo object contains the language of the active user.

Field value: The contents of the field to be checked. In addition to fixed values, regular expressions (see below) can also be entered here. If the regular expression matches, the condition is considered true.

Regular expression: If enabled, *Field value* is treated as a regular expression (RegEx) and applied to the content of the respective field according to the JavaScript method *String.prototype.match*. If the expression is found at any point, the condition is considered true. The regular expression is instantiated without flags; use of ignore case must be reflected in the expression itself if necessary.

Instance: In this field, you can enter an instance other than the default instance. If you do not enter anything, the default instance is applied. The instance entered here must be set under *General*.

SAP system: The name of the SAP system that ELO Smart Link connects to, if the rule applies.

Configuration – searchHelps.config – search help

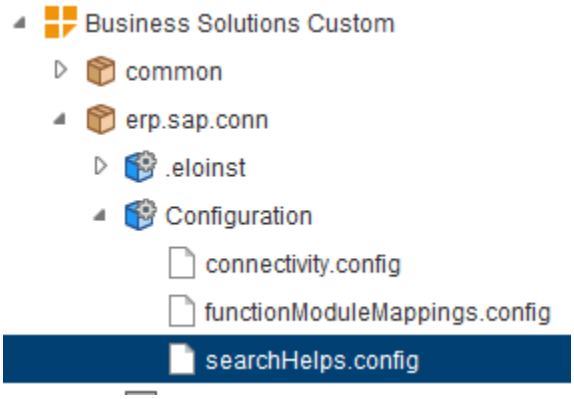

Fig.: ELO Business Solutions Custom, erp.sap.conn, 'Configuration' folder

Once you have implemented the *erp.sap.conn* package, you can go to *Configuration* > *searchHelps.config*.

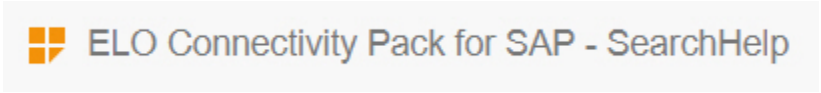

Fig.: Header in the configuration interface

The ELO Connectivity Pack – Datatransfer configuration interface opens in ELO.

Please note

Only edit the configuration in the *Business Solutions Custom* path to ensure the system can be updated in the future. When a module is updated, the configuration in the Business Solutions path can be overwritten.

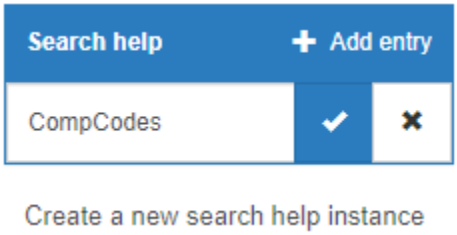

Fig.: Configuration interface, creating a search help

This configuration interface only has one area where you can create and configure search helps.

Information

When installing on ELO 10, you may not be able to open the configuration interface. For more information, refer to the Installation information.

Information

You can create a search help with *Add entry*. You can choose any name for the search help, but it is recommended that you use only upper and lower case letters of the Latin alphabet.

Search help

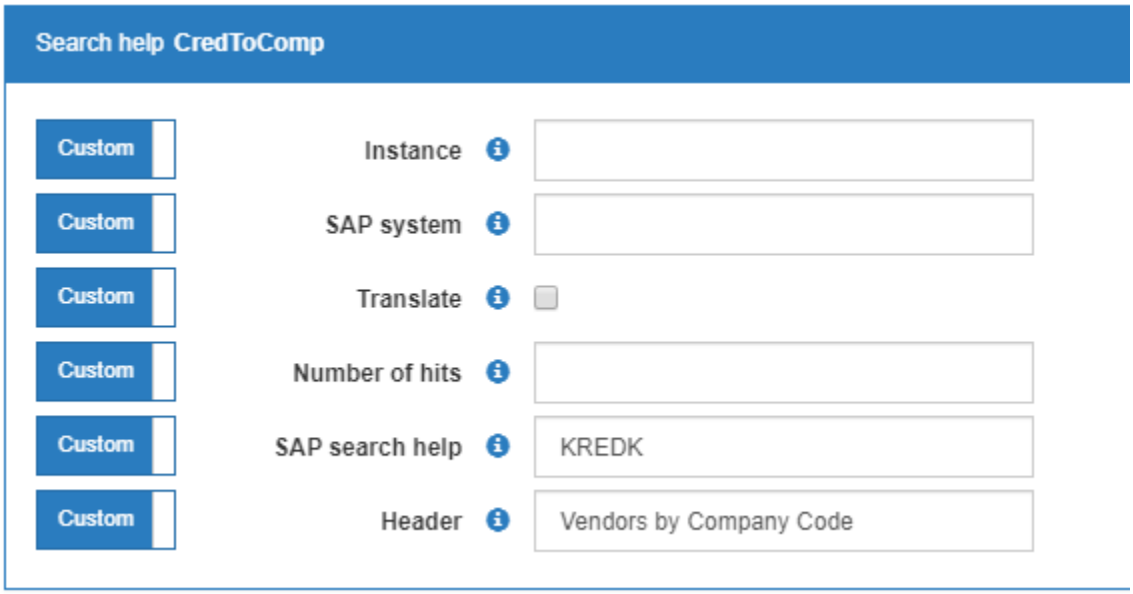

Fig.: Configuration interface, loading a search help

Instance: The name of a configured ELO Smart Link instance. To use the default instance from the *connectivity.conf* file or one of the context-dependent options, this field can be left empty.

SAP system: The system ID of the configured SAP system in the ELO Smart Link configuration. To use the default instance, this field can be left blank.

Translate: If selected, the texts from the header are treated as translation variables. Also refer to the section Use translation variables in search helps.

Number of hits: Specifies the number of hits displayed in the search help. If you do not specify a value here, the number of hits is limited to 50. A maximum of 1000 hits is allowed. For performance reasons, you should set a lower value.

SAP search help: The technical identifier of the SAP search help that is addressed.

Title: The title or translation variable used for the SAP search help.

> **Information**
>
> It is only possible to address elementary SAP search helps. If you want to use customer-specific search helps, make sure to create them as elementary search helps.

When you click the *Load* button, the headers of the stored search help in the SAP system are queried. This process either addresses the instance and the SAP system from the *connectivity.config* file, or if values are maintained for the instance and the SAP system in the search help itself, the instance or SAP system is addressed.

> **Information**
>
> If the contents of the table remain empty after clicking the *Load* button, this can have various causes. In each case, check the connection parameters in the *connectivity.config* file, the settings in ELO Smart Link, and check that the required ELO Business Solutions version is installed. Another cause could be that there are missing permissions in the respective SAP system.

Headers

After clicking the *Load* button, the headers of the stored SAP search help were loaded. This screenshot shows the headers for the basic SAP search help *KREDK* (vendors for each company code).

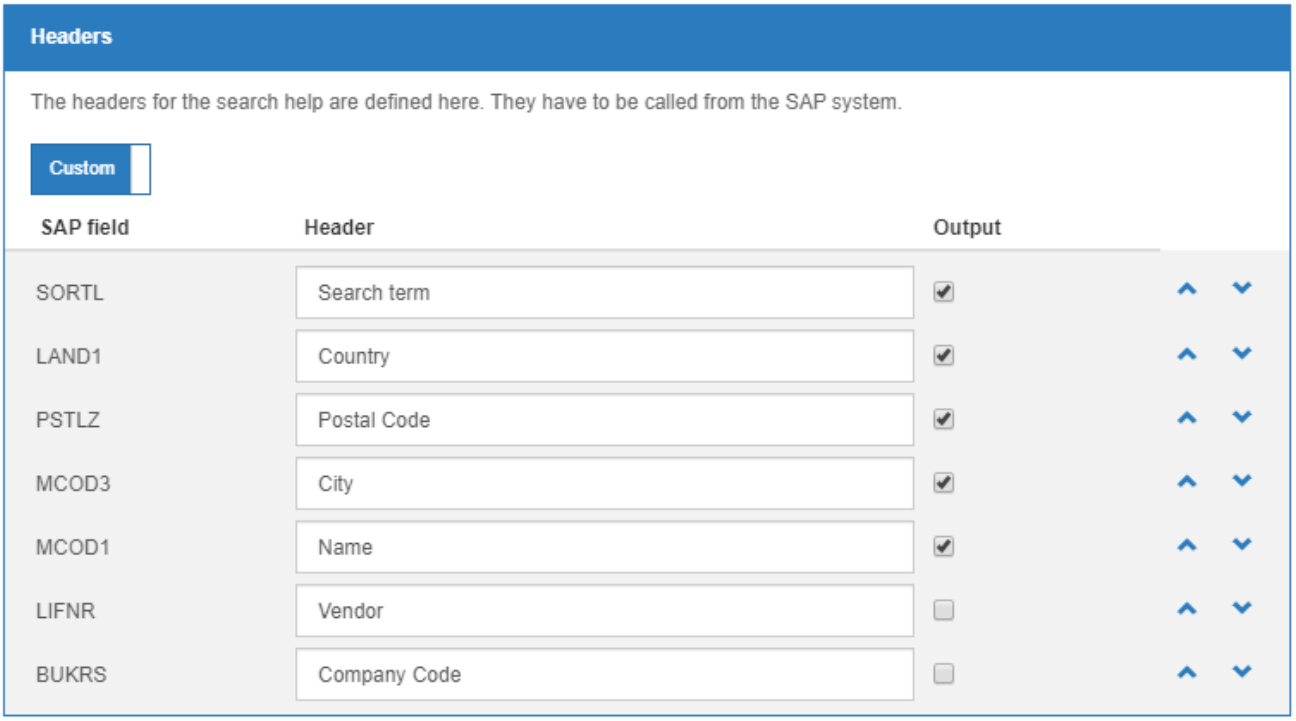

Fig.: Configuration interface, loaded headers

SAP field: Name of the SAP fields whose values are used to output the search help.

Title in header: The title used in the header of the search help. This title is displayed in the header of the dynamic search helps in the form. You can use a translation variable.

Output: If enabled, the field is output in the search help.

You can change the order of the columns using the arrows at the far right.

Output

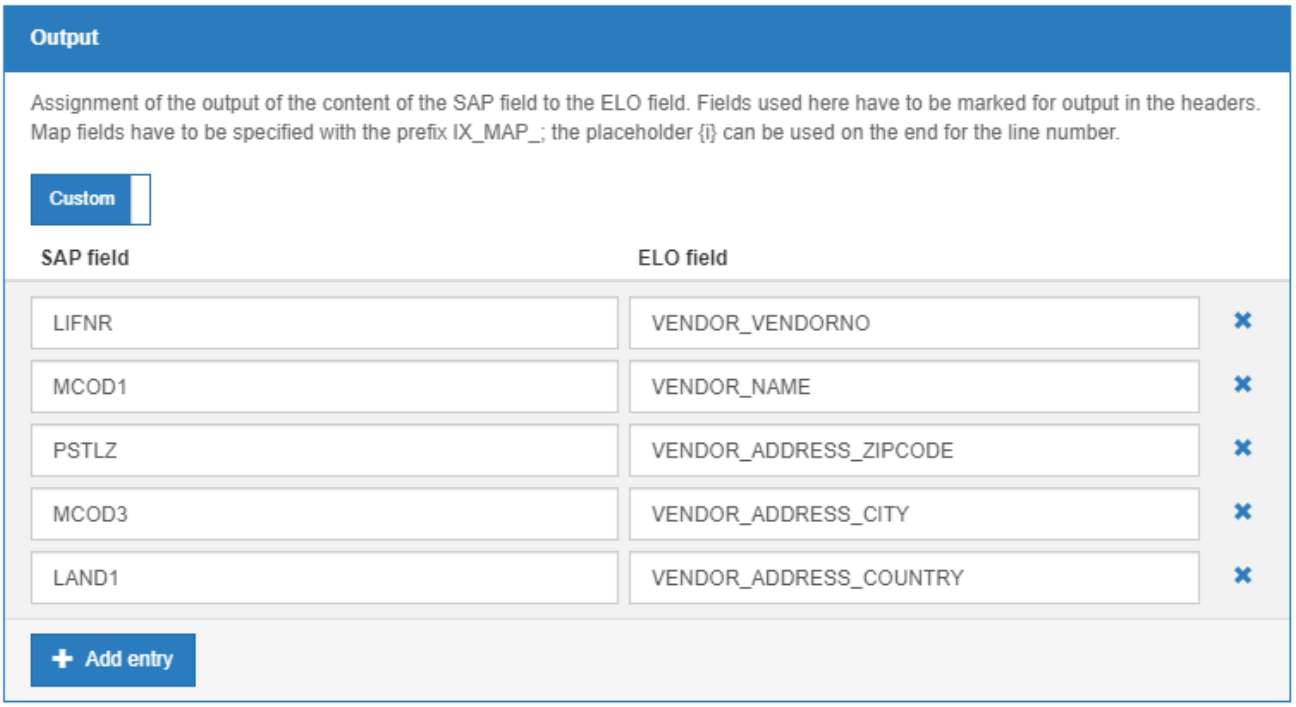

Fig.: Configuration interface – search help output

This configuration area is where you define the mapping of the output of the content of the SAP field to the ELO fields.

SAP fields used here have to be marked for output in the *Headers* area. MAP fields must be preceded by the prefix `IX_MAP_` and the placeholder `{i}` can be used on the end for the line number.

> **Information**
>
> Make sure you enter the correct information in the SAP fields and ELO fields. The field names will not be checked at this point.

Search

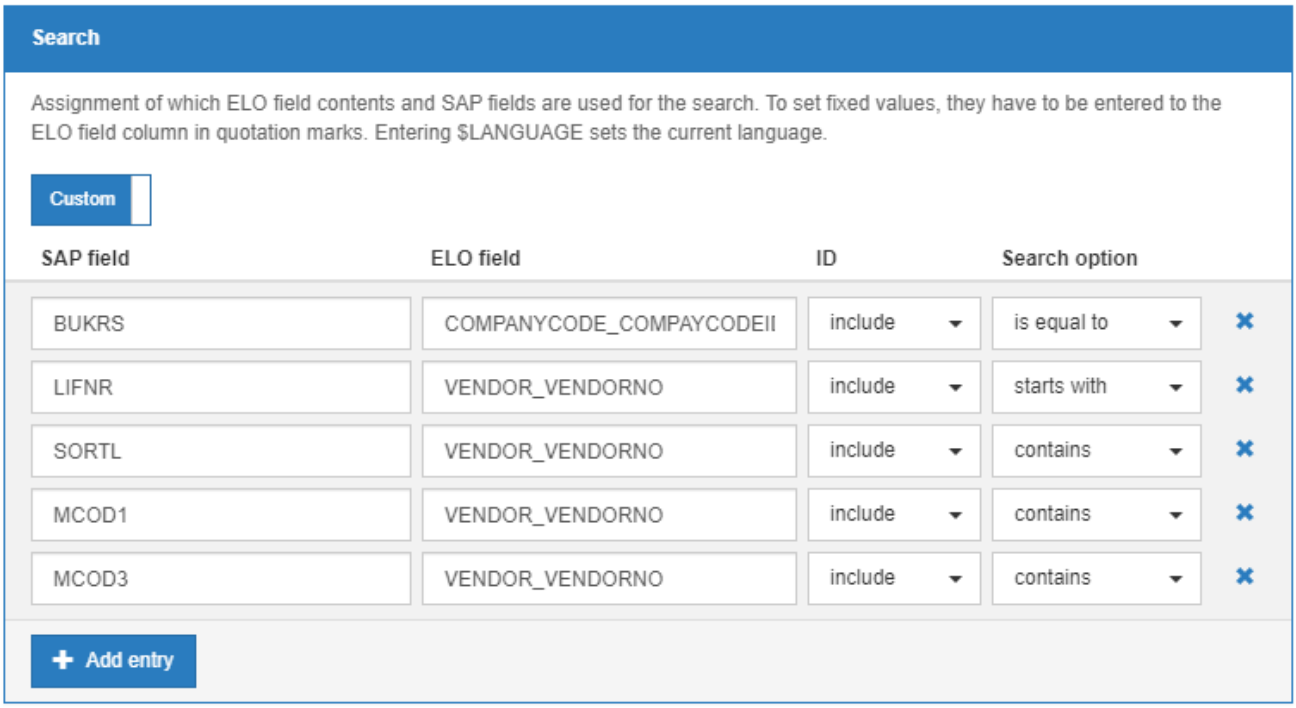

Fig.: Configuration interface, search

The ELO field contents are assigned to the corresponding SAP fields as shown in the figure *Configuration interface, search*. This is where you specify which criteria to apply to search in the SAP fields.

SAP field: Name of the SAP field being searched.

ELO field: The value from this field is used for the search in the SAP field. To set fixed values, they have to be entered into the *ELO field* column in quotation marks (").

Fig.: Example of a search for a fixed value: "VST"

In this example, the fixed value `"VST"` is transferred for the transaction key `KTOSL` in the search help for tax codes.

Fig.: Configuring a search for a system-dependent language

Entering `$LANGUAGE` sets the current language.

Fig.: Example of a search via MAP fields

When using map fields, the prefix `IX_MAP_` must be placed at the start of the map field, or the suffix `{i}` must be attached to map fields with counters.

ID: You can use an ID to define the search in the SAP system in detail. There are three options available:

- include
- exclude
- *or* include

ID	Explanation
include	Mandatory criterion: All entries must match the criteria. All other entries are excluded from the results.
exclude	Exclude criterion: Entries that match the criteria are explicitly excluded from the results.
or include	Optional criterion: Unlike *include*, matching entries are included in the results without explicitly excluding the remaining entries. This ID is normally used if several criteria can apply to a field value.

The following example illustrates how these ID can be used. We need the search to return purchase order data for specific company codes (BUKRS) and plants (WERKS).

Example 1:

include – BUKRS = 1000

or include – WERKS = 1000

or include – WERKS = 2000

With this configuration, you obtain all results (purchase orders) created in company code 1000 and in plant 1000 or 2000. The company code is taken into account in both *"or"* cases.

Example 2:

include – BUKRS = 1000

exclude – WERKS = 3000

With this configuration, you obtain all results (purchase orders) created in company code 1000, but none from plant 3000.

Search option: At this point you can choose from four options:

- Is equal to: The search result must match the respective value exactly.
- Contains: The search result contains part of the respective value. Example: The account name *Other payables* contains the text `payables`. Exact matches also meet the criteria.
- Starts with: The search result must begin with the respective value.
- Ends with: The search result must end with the respective value.

You now need to complete a few steps before you can use the search help in ELO. First, you need to create an IX script file, which you can find in the Invoice SAP template provided by ELO under the following path, for example:

Administration//Business Solutions//erp.sap.invoice//IndexServer Scripting Base//DynKwl > erp.sap.invoice.ix.dynkwl.CompCode

This file refers to the created configuration and the entry. This file and the search help you created can now be used in fields ((GRP) in the ELO Administration Console) or forms.

Use translation variables in search helps

If the *Translate* check box was set in the configuration of the search helps, the texts from the header are treated as translation variables. For more information on translation variables, see the ELO workflow documentation.

This documentation describes the search help *Company Codes – CompCodes (H_T001)* as an example.

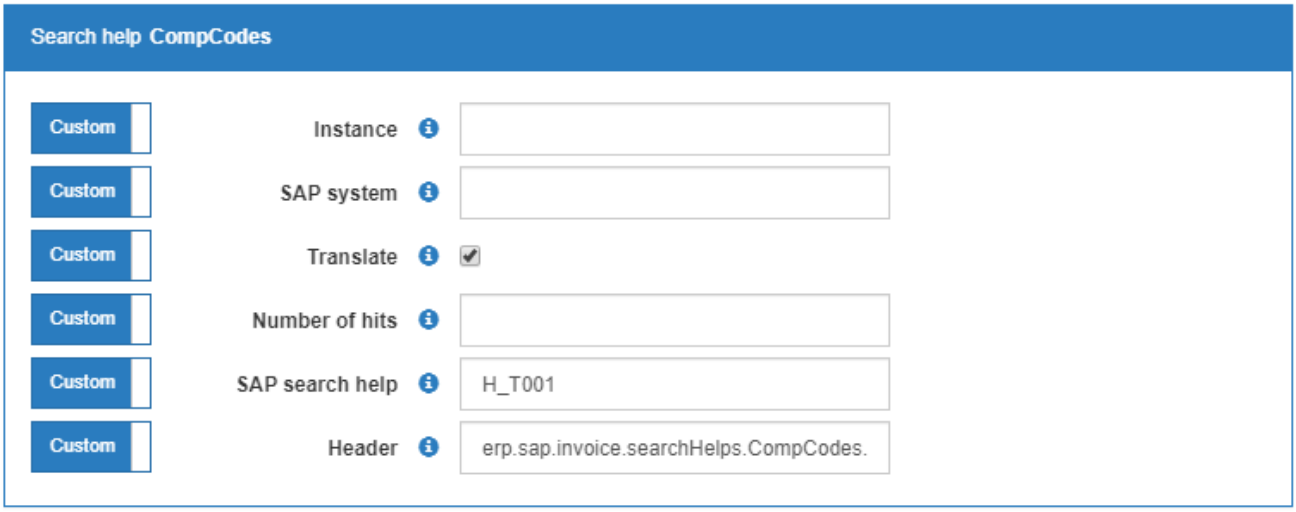

Fig.: Search help translation variables, title

Field	Value
Instance	-
SAP system	-
Translate	enabled
Number of hits	-
SAP search help	H_T001
Title	erp.sap.invoice.searchHelps.CompCodes.

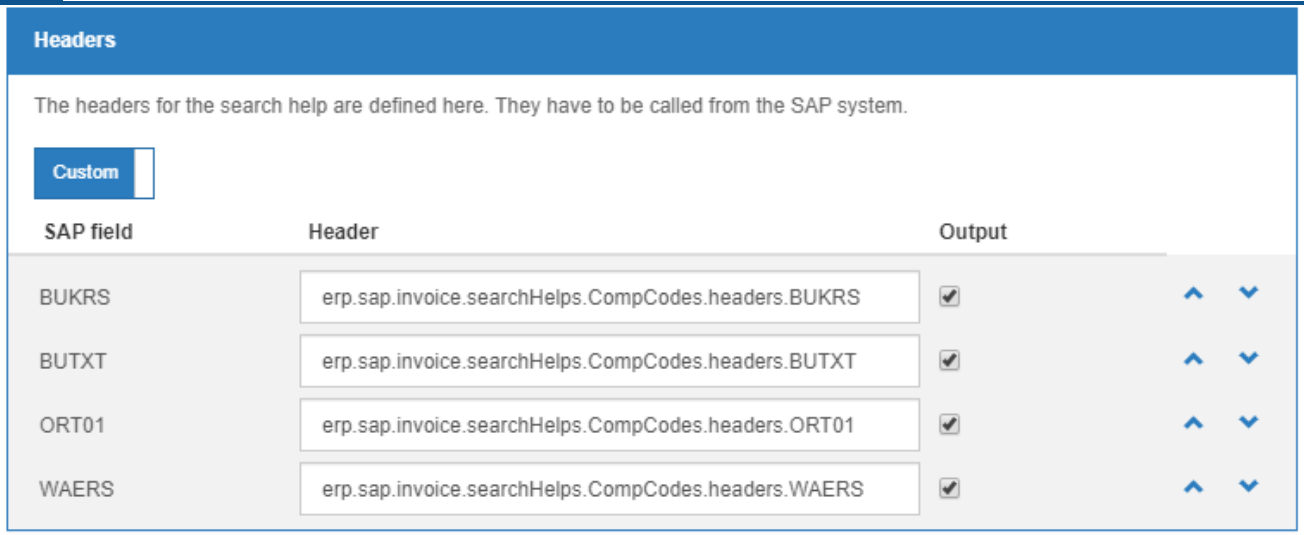

Fig.: Search help translation variables, headers

SAP field	Title in header	Output
BUKRS	erp.sap.invoice.searchHelps.CompCodes.headers.BUKRS	Yes
BUTXT	erp.sap.invoice.searchHelps.CompCodes.headers.BUTXT	Yes
ORT01	erp.sap.invoice.searchHelps.CompCodes.headers.ORT01	Yes
WAERS	erp.sap.invoice.searchHelps.CompCodes.headers.WAERS	Yes

Configuration – Function module mapping

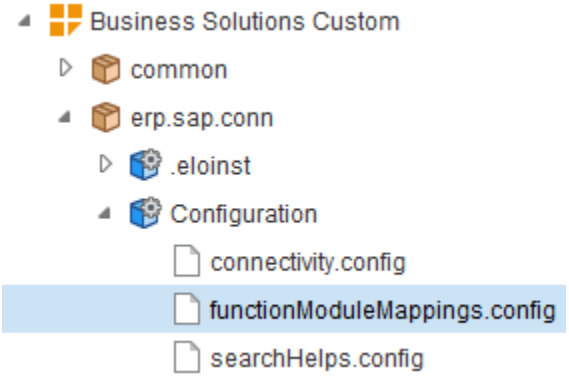

Fig.: ELO Business Solutions Custom, erp.sap.conn, 'Configuration' folder

Once you have implemented the *erp.sap.conn* package, you can go to *Configuration* > *functionModuleMappings.config*. The ELO Connectivity Pack – Datatransfer configuration interface opens in ELO.

Please note

Only edit the configuration in the *Business Solutions Custom* path to ensure the system can be updated in the future. When a module is updated, the configuration in the *Business Solutions* path can be overwritten.

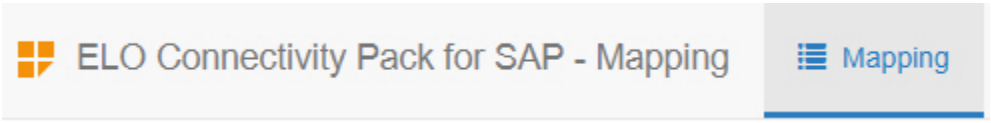

Fig.: Header in the configuration interface

In this configuration interface, function modules are mapped by assigning the SAP fields and SAP tables to the corresponding ELO fields and ELO map fields or vice versa (export or import parameters).

Information

When installing the solution, you may not be able to open the configuration interface. For more information, refer to the Installation information.

Create function module mapping

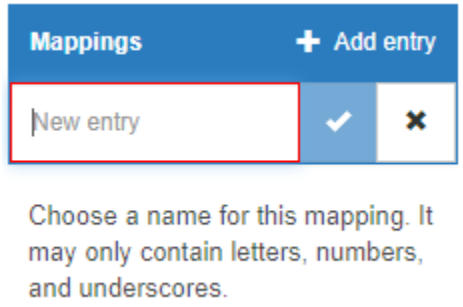

Fig.: Configuration interface, creating a mapping configuration

You need to create a new mapping for each function module. The name can only consist of letters, numerals and the underscore. This name is used later, for example, when implementing in workflow nodes.

The configuration options are outlined in the next step.

Information

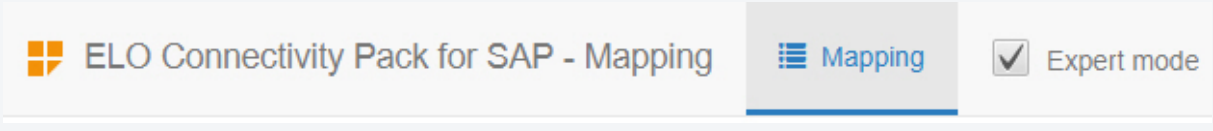

Fig.: Configuration interface, expert mode enabled

Fields marked with (*) in the explanation are only visible in expert mode. You can enable this mode using the check box next to *Mapping* at the top of the configuration interface.

Mapping

Fig.: Configuration interface, configuring a function module

Instance: The name of a configured ELO Smart Link instance or context-dependent options. To use the default instance defined in section connectivity.config – General, you can leave the field empty.

SAP system: The system ID of the configured SAP system in the ELO Smart Link configuration. You can leave this field blank to use the default SAP system.

Function module: The name of the function module in SAP for which the mapping is being performed. It must be RFC-enabled.

Commit: Only a database commit writes the data created or changed by a transactional function module to the database. Not every function module performs the commit. In this case, you need to enable the option to explicitly perform a commit. If in doubt, refer to the SAP documentation for the function module in question to find out whether it performs a commit.

Throw BAPIRET error: If the exporting or tables parameters contain the data type *BAPIRET*, the system checks for errors and forwards them. This function simplifies error handling.

Queue: If this option is enabled, execution of the function module is processed via the ELO Smart Link queue. This can reduce the load on the system for mappings that are executed frequently and makes it easier to trace Q entries.

Prepare RF for response (*): Enter a registered ELO Indexserver function to process the response. Normally, the supplied function `RF_erp_sap_conn_service_WriteBack` will suffice.

Preprocessing IX function (*): An ELO Indexserver function that returns an object with data, for example to perform conversions or to collect data if these actions cannot be implemented with mapping alone. The function gets the object ID and must return the desired object as a JSON string. It is then available under the property *preObj* in Handlebars syntax.

Overwrite fields: If this option is enabled, the content of mapped target fields is overwritten. If the check box is not selected and an ELO field contains another entry, this is automatically entered in the next line of the metadata field.

Field for error messages: Possible error messages, such as posting error messages or BAPIRET error messages, are entered in this field.

Error message field type: This is where you define the field type of the error message. This can be either a field (GRP) or a map field.

Status field: Name of the field assigned the following success/error value after execution. This is helpful when monitoring workflows, for example.

Success value status field: The value written to the configured status field in case of success.

Error value status field: The value written to the configured status field in case of an error.

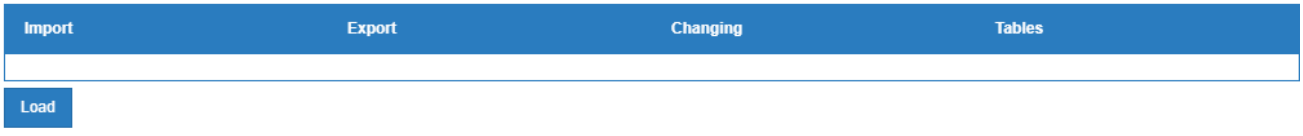

Fig.: 'Load' button in the configuration interface

If the required configuration entries have been maintained and the correct function module has been stored, you can click the *Load* button to load and display the required parameters of the function module.

Information

It may take some time to load all fields. You cannot save any changes during this time. You should only continue processing once all fields have been successfully loaded.

This example can be found in the Invoice SAP template as *invoicePostFI*.

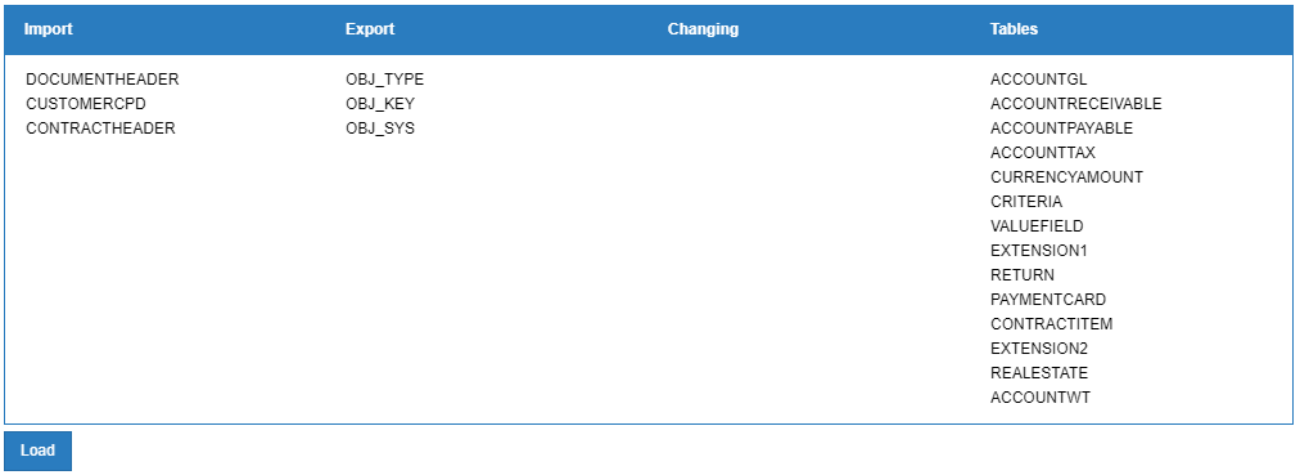

Fig.: Completed configuration interface after successful loading

Information

If the contents of the table remain empty after clicking the *Load* button, this can have various causes. In each case, check the connection parameters in the *connectivity.config* file, the settings in ELO Smart Link, and check that the required ELO Business Solutions version is installed. Another cause could be that there are missing permissions in the respective SAP system.

Clicking on the individual parameters (e.g. *DOCUMENTHEADER*) opens the configuration for mapping the selected entry.

The three types used for communication with the interface parameter types and that build on each other are explained below:

Elementary type: This type is a simple value and behaves like a single field in a table.

Structure: In the SAP environment, a (flat) structure is a data type that is comparable to a single row in a database table. A structure can contain any number of columns. In this context, we also refer to fields instead of columns.

Table: Unlike structures, table types can contain any number of rows. A table type always has a structure as a line type.

Interface parameter types

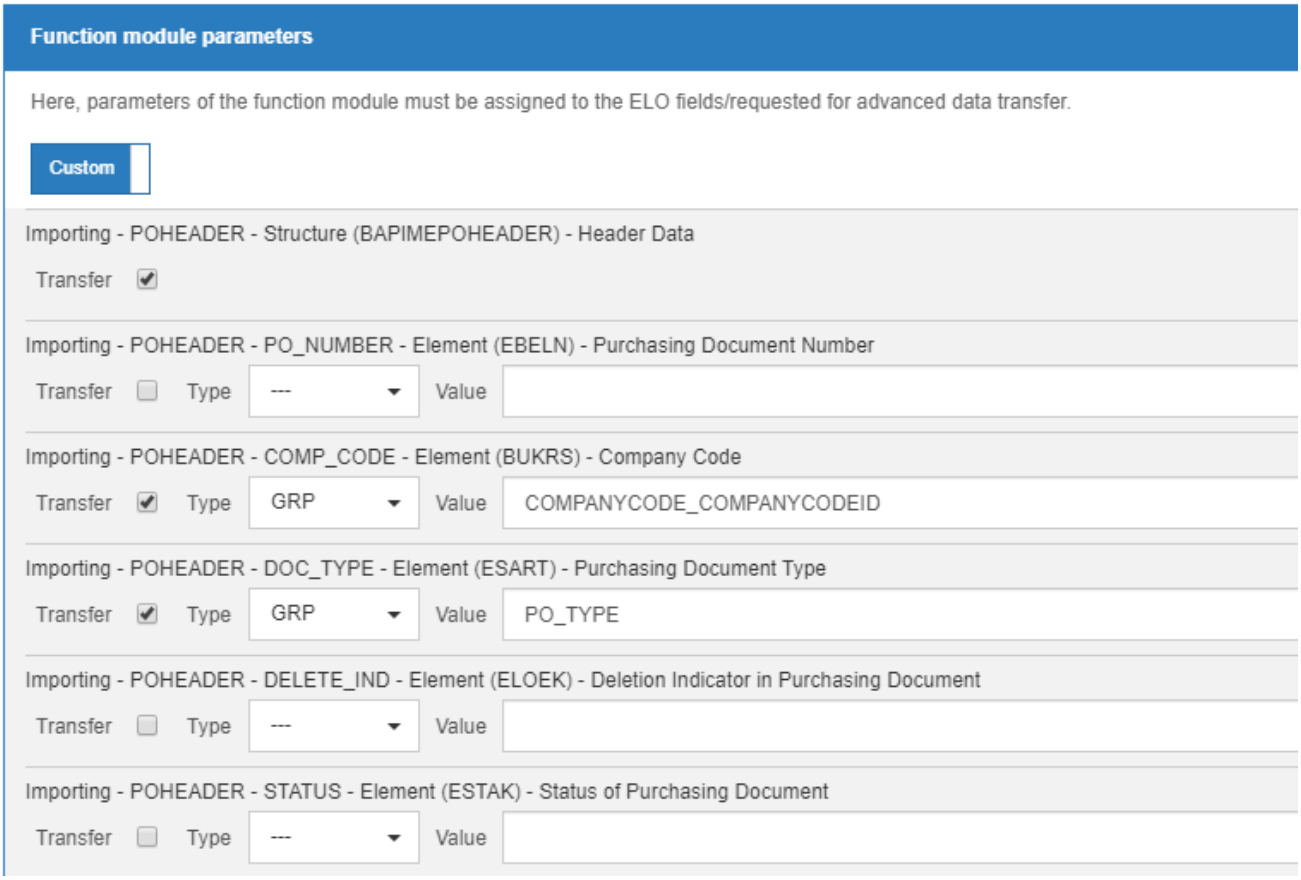

Fig.: Configuration interface, mapping

There are four types of interface parameters that can be used:

Import: An import parameter can only be passed to the function module. All three types can be passed here: elementary types, structures, and tables.

Export: The contents of an export parameter are available if the function module ended without errors. Export parameters are only returned to ELO. You can use the same types as for import.

Changing: These parameters are both import and export parameters and, like these, can also consist of all three types. A changing parameter can therefore be transferred and requested. Before the function module is called, the changing parameter behaves like an import parameter. When the call is complete, the changing parameter acts as an export parameter.

Tables: This parameter only allows tables as types. Like changing parameters, these can be transferred and requested.

After selecting a specific parameter, the components corresponding to the type appear, which can now be assigned to the ELO fields or requested for the advanced data transfer.

Transfer/Request: If this field is enabled, the values are transferred to the corresponding fields.

If *Transfer* was selected, values are transferred to the SAP system. With *Request*, the SAP system transfers the data to ELO.

You can only enable *Transfer* for import parameters. *Request* is only allowed for export parameters.

Both actions are allowed for tables. In the SAP context, however, it is common practice to use tables either as an import or as an export.

Type: The ELO field type. As tables consist of structures, and structures consist of elements and counting structure elements, the types that can be selected depend only on whether the end result is an element or a counting structure element. The type determines what you can enter in the *Value* field. This is explained below.

There are two types of counting structure elements:

- MAPLOOP: The value entered represents an ELO map field, which automatically runs through a loop. If you enter a map field name such as *POS* here, the function assumes that a map field with the name *POS* exists. In this case, the number following *POS* counts upwards until the map field with the name no longer exists. Elements that follow a counting structure element can have the entry *{I}*, for which the number of the loop iteration is then used. For example, for a map field with the name *VALUE1 VALUE2* etc., you only need to set VALUE{I}.
- OBJ_PROP_LOOP: This counter structure element is used to transfer a self-defined object. For a detailed explanation, see Preprocessing IX function in the section *Create function module mapping* You can see the structure of the JSON object that is passed in the script *erp.sap.invoice.ix.GetPostingLines*. preObj is always the object returned by the IX function for preprocessing. This object can have any number of attributes that can be addressed using Handlebars syntax. An example of this is the *invoicePostFI* function in the *erp.sap.Invoice* (Invoice SAP template) package. Here, preObj.accountgl is specified as the counter structure element for the table `ACCOUNTGL`. Therefore, `preObj.accountgl` provides access to multiple attributes, which are now used for iteration. All objects that have been passed through are transferred to the subsequent elements via Handlebars. For the element `ACCOUNTGL-ITEMNO_ACC`, for example, the entry defined with the following expression is transferred:

```
{{{preObj.accountgl.{I}.itemno_acc}}}
```

The field type must be *HTMPL*. With `preObj.accountgl`, you address the array of objects, the following entry `{I}` defines the index of the current loop iteration, and `itemno_acc` points to the attribute of the object, which is ultimately transferred.

There are four types of elements that can be transferred. In principle, it is possible to transfer elements in both directions (ELO to SAP and SAP to ELO). The transfer direction from ELO to the SAP system is described below.

- GRP: This is where you define the name of the field from the ELO metadata form from which the value is read.
- HTMPL: HTMPL stands for Handlebars template (see section Handlebars syntax).
- MAP: An ELO map field is used as the source.
-

FIX: The value stored here is transferred. The field can also contain an `{I}` in the loop context.
Value: The value that is entered here depends on the selected types. Possible entries are already defined under *Type*.

> **Information**
>
> The placeholder `{i}` can be used on the end for the line number, in this case with a capital "I".

Advanced data transfer

The next area of the configuration interface is reserved for advanced data transfer from the SAP system. At this point, you have two possible options for advanced data transfer.

Handlebars syntax

Handlebars is a template engine that dynamically generates character strings. It uses a special syntax to manipulate and convert input values, sometimes with the help of conditional branching (*"if/else"*). An output rule always consists of a character string, known as the Handlebars string. Official information on Handlebars syntax can be found on the website handlebarsjs.com.

In addition to the standard Handlebars functionality, there are some helper functions available.

> **Information**
>
> You will find a list of the available helper functions on the ELO SupportWeb in the documentation on the class *sol.common.Template* in the *Business Solutions Common* package.

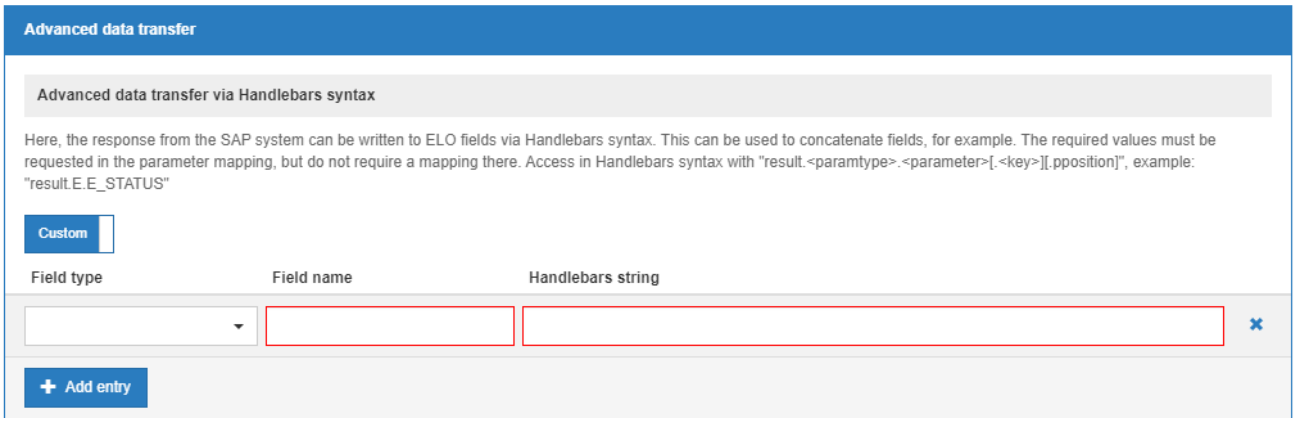

Fig.: Configuration interface, Handlebars

Handlebars syntax allows you to write the response from the SAP system to ELO fields. This can be used to concatenate or split fields, for example. The required values must be requested in the parameter mapping, but do not require a mapping there.

An example of advanced data transfer using Handlebars syntax:

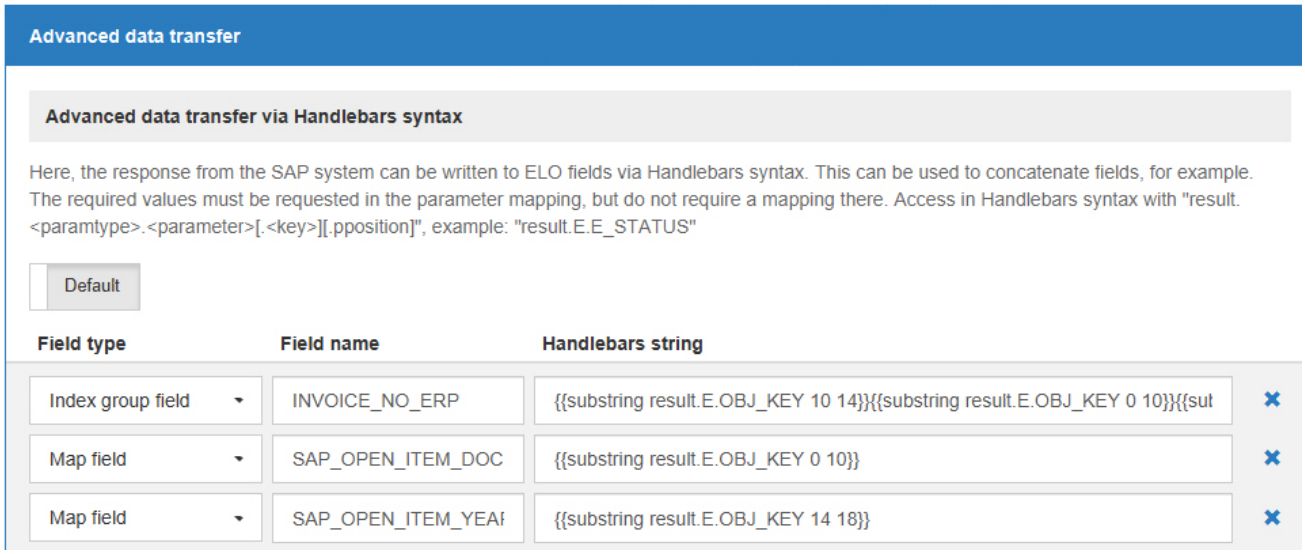

Fig.: Configuration interface, Handlebars syntax

- Field name: `INVOICE_NO_ERP`
- Field type: ELO field
- Handlebars string:

```
{{substring result.E.OBJ_KEY 10 14}}
{{substring result.E.OBJ_KEY 0 10}}
{{substring result.E.OBJ_KEY 14 18}}
```

- Field name: `SAP_OPEN_ITEM_DOC`
- Field type: MAP field
- Handlebars string:

```
{{substring result.E.OBJ_KEY 0 10}}
```

- Field name: `SAP_OPEN_ITEM_YEAR`
- Field type: MAP field
- Handlebars string:

```
{{substring result.E.OBJ_KEY 14 18}}
```

In the above example, a response from the SAP system is converted, addressing the *result* object.

Handlebars can also be used to transfer data from ELO to the SAP system. ELO fields can be addressed with the convention *sord.objKeys.<FIELDNAME>*, whereas ELO map fields can be addressed with *sord.mapKeys.<FIELDNAME>*. For more information, refer to the Business Solutions documentation.

Access to the *result* object is structured as follows:

```
result.<paramtype>.<parameter>[.<key>][.pposition]
```

`<paramtype>` describes the parameter, in this case `E` for Export, `C` for Changing, and`T` for Table.

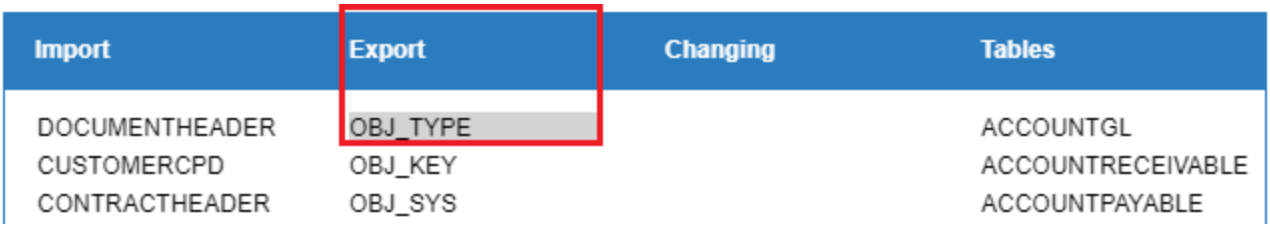

Fig.: Advanced data transfer of the result object

`<parameter>` corresponds to the name of the parameter of the SAP function module. In the above example, it would be the export parameter with the name `OBJ_TYPE`.

`[.<key>]` is only relevant if the parameter is either a structure or a table type. The field is addressed according to the field name.

`[.pposition]` is only relevant for table types. You can address a line of the table selected using `<parameter>`. Note that you must place the position in square brackets. The example for addressing the second line of a table looks like this:

```
result.T.TABLE.FIELD.[2]
```

SAP table values in MAP tables

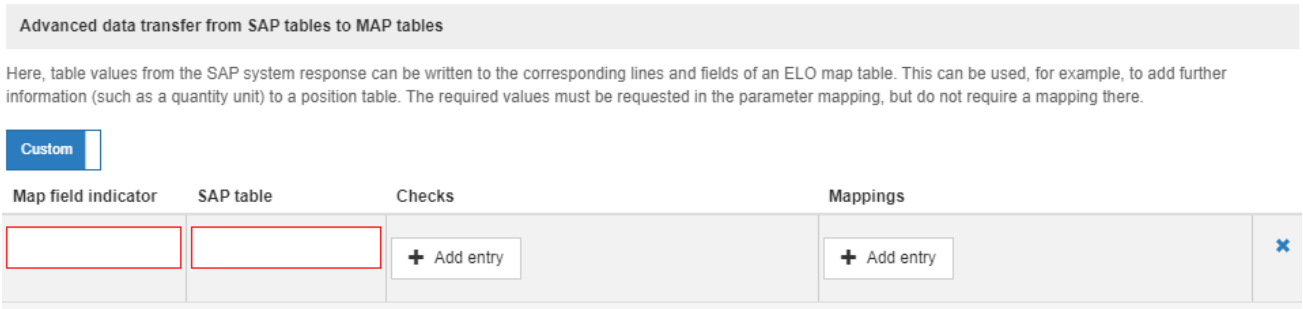

Fig.: Configuration interface, SAP table values in MAP tables

In addition to advanced data transfer using Handlebars, this configuration option also allows you to write table data from the SAP system with conditions to the corresponding lines and fields of an ELO MAP table.

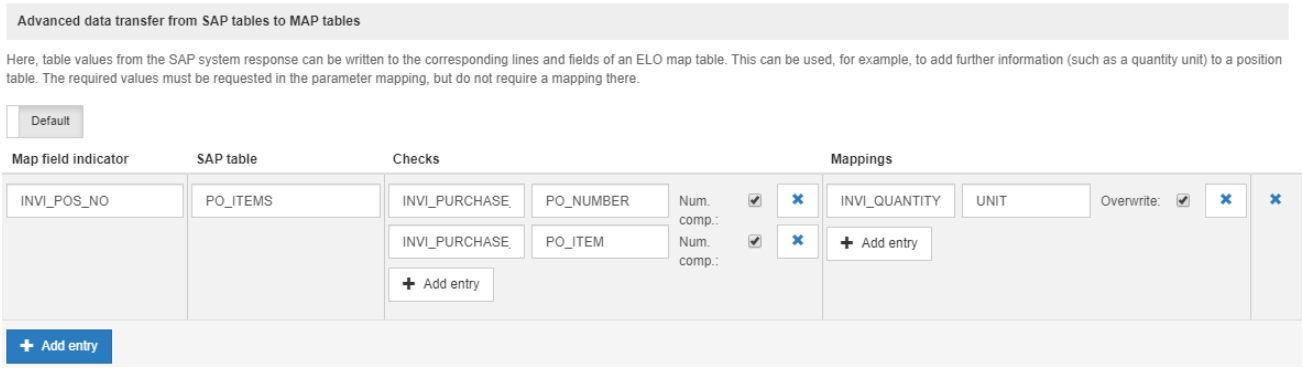

Fig.: Configuration interface, SAP table values in MAP tables (example)

In the example shown above, the loop passes through the ELO map field `INVI_POS_NO1` to `INVI_POS_NOX` (last available number; these iterations are marked with `{i}`).

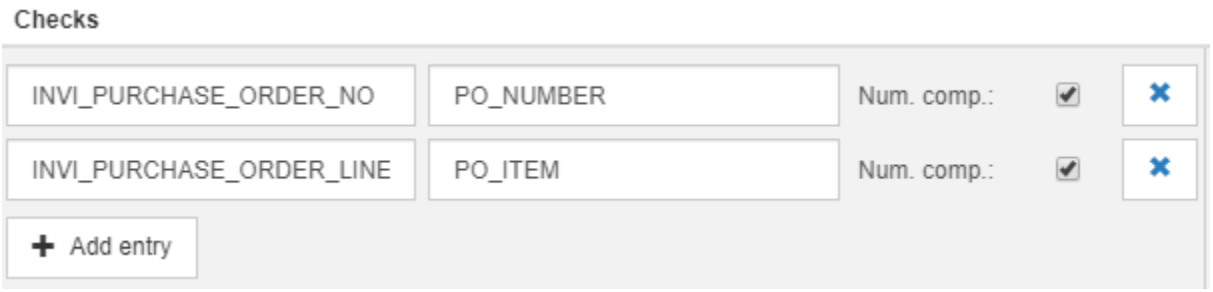

Fig.: Configuration interface, SAP table values in MAP tables, 'Checks'

At the same time, the function passes through all rows in the SAP table `PO_ITEMS` and checks for matches (see figure above) between the SAP table field `PO_ITEMS-PO_NUMBER` and the ELO map field `INVI_PURCHASE_ORDER_NO{i}`, as well as between the SAP table field `PO_ITEMS-PO_ITEM` and the ELO map field `INVI_PURCHASE_ORDER_LINE_NO{i}`.

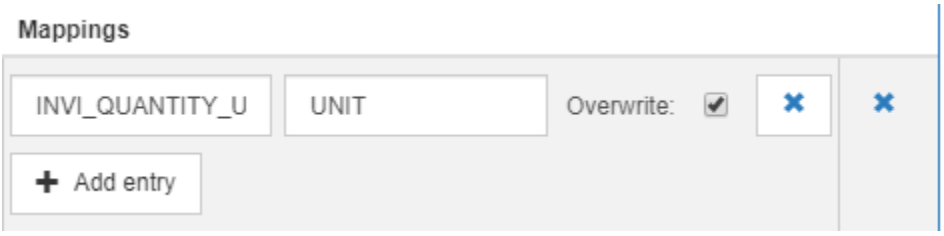

Fig.: Configuration interface, SAP table values in MAP tables, 'Mappings'

If this is the case, the data is mapped from the SAP table field `PO_ITEMS-UNIT` to the ELO map field `INVI_QUANTITY_UNIT{i}`.

Process and implementation examples

This chapter explains how to implement the functions provided in ELO Connectivity Pack – Datatransfer.

Search help

Follow the instructions in the chapter Function module mapping – Function module. Once you have created a new search help, you need to create a search help script.

Example

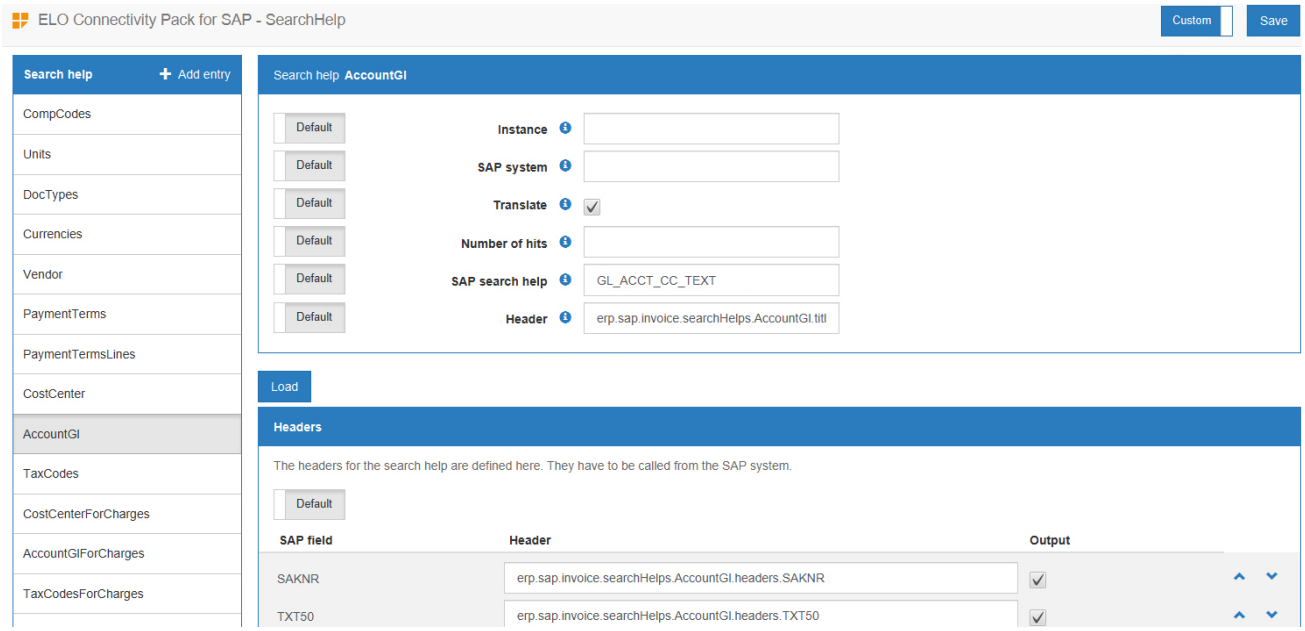

Fig.: Implementation example search help 'AccountGl'

The search help *AccountGl* (general ledger accounts) is defined as follows:

File: erp.sap.invoice.ix.dynkwl.AccountGl

```
importPackage(Packages.de.elo.ix.jscript)
importPackage(Packages.de.elo.ix.scripting)
//@include lib_Class.js
//@include lib_erp.sap.conn.ix.GenericSearchHelp.js
var configPath = "/erp.sap.invoice/Configuration/searchHelps.config",
  kwlName = "AccountGl"
```

This example refers to the SAP search help `GL_ACCT_CC_TEXT`, which was created in the configuration with the search help name *AccountGl*. Use this name in the script file.

In the next step, you need to integrate the search help into the ELO form.

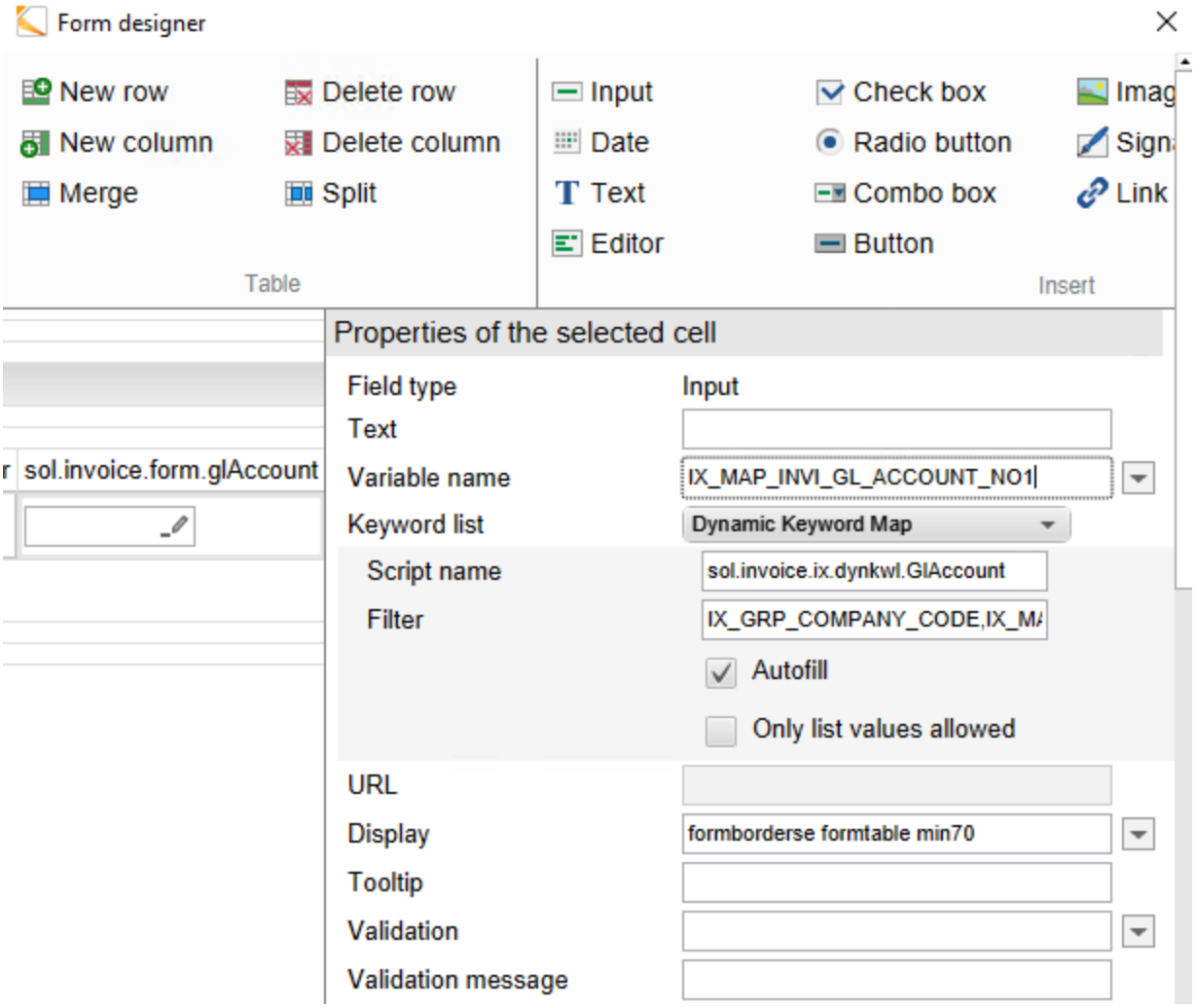

Fig.: Usage example for a search help

It is important that you select *Dynamic Keyword Map* as the keyword list. Under *Script name*, you must enter the Indexserver script created for your own search help.

In the example, this is the script name `erp.sap.invoice.ix.dynkwl.AccountGl`. The value `IX_GRP_COMPANY_CODE,IX_MAP_INVI_GL_ACCOUNT_NO{i}` is specified for the filter.

Function module mapping – Function module

Function modules can be called either by script or via a workflow node. First, configure a function module as described in section Configuration – Function module mapping. The images illustrate the example.

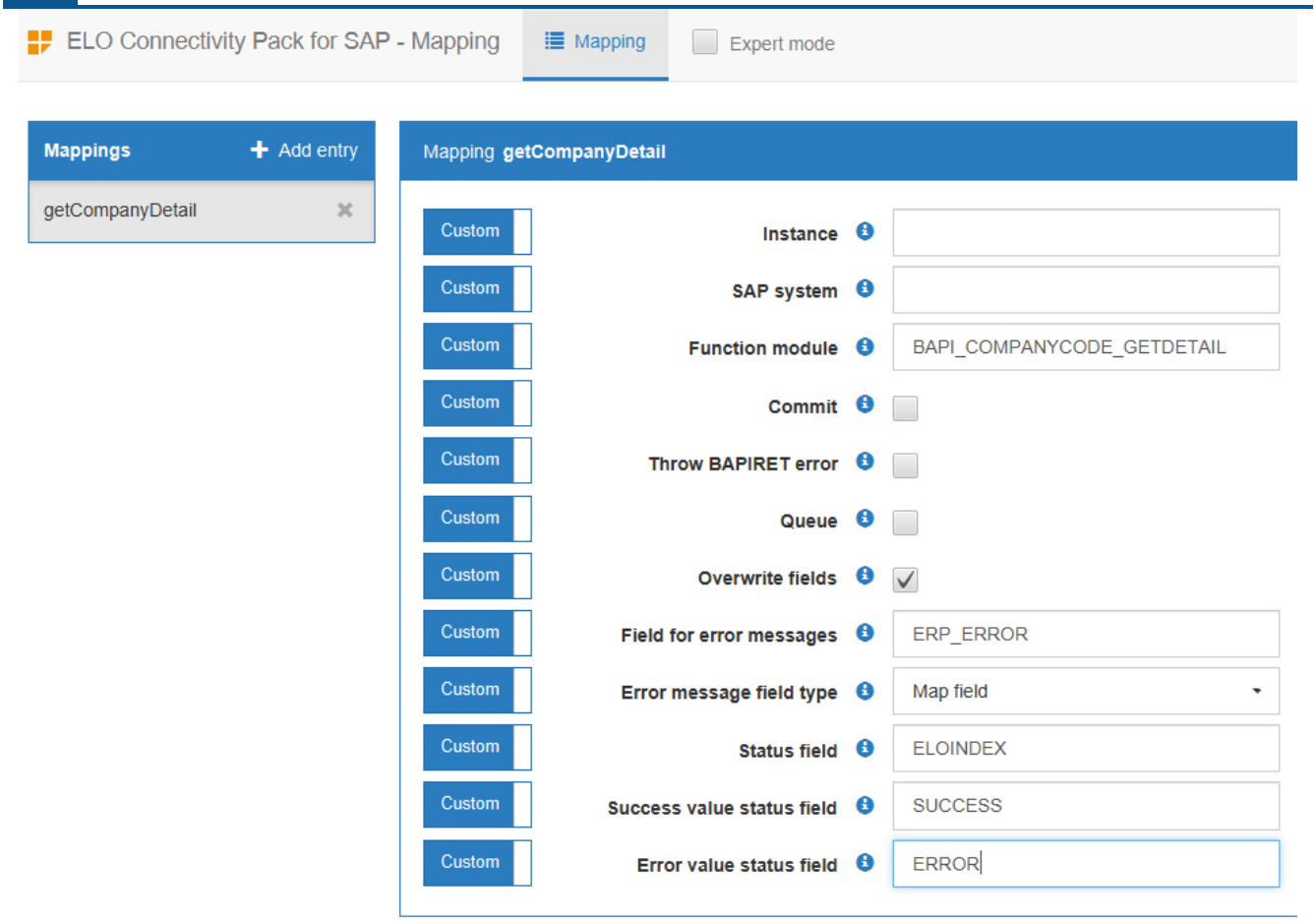

Fig.: 'Configure function module' example – Settings for the 'getCompanyDetail' mapping

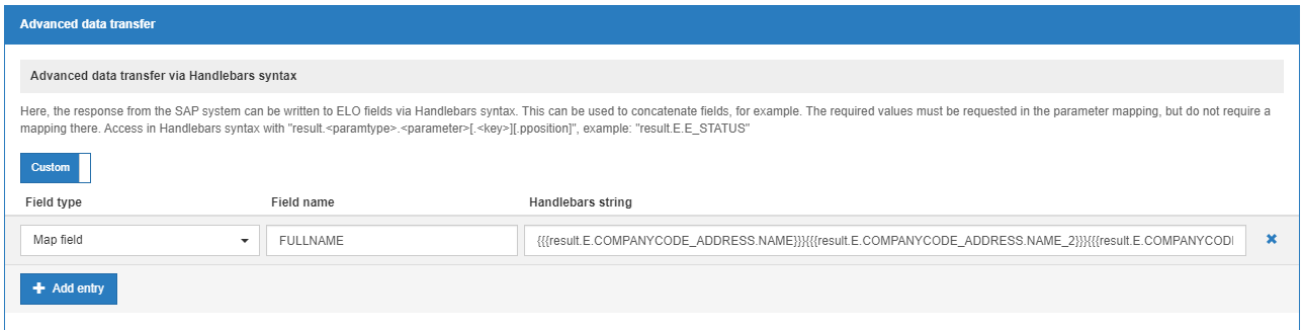

Fig.: 'Configure function module' example – Settings for the 'Advanced data transfer' area

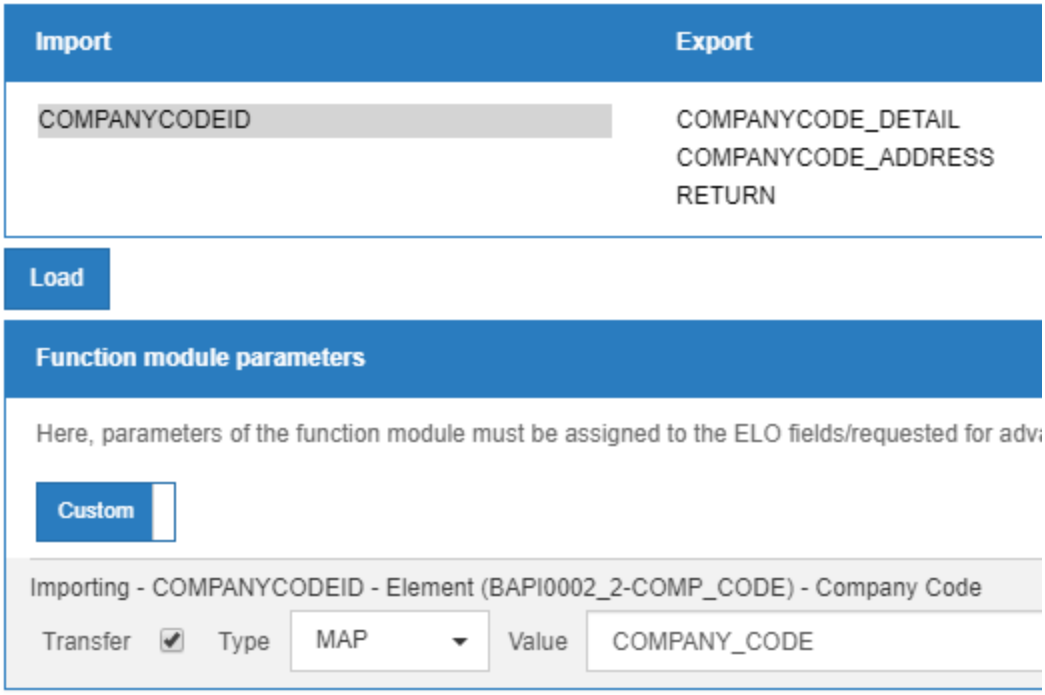

Fig.: 'Configure function module' example – Parameters of the 'COMPANYCODEID' import function module

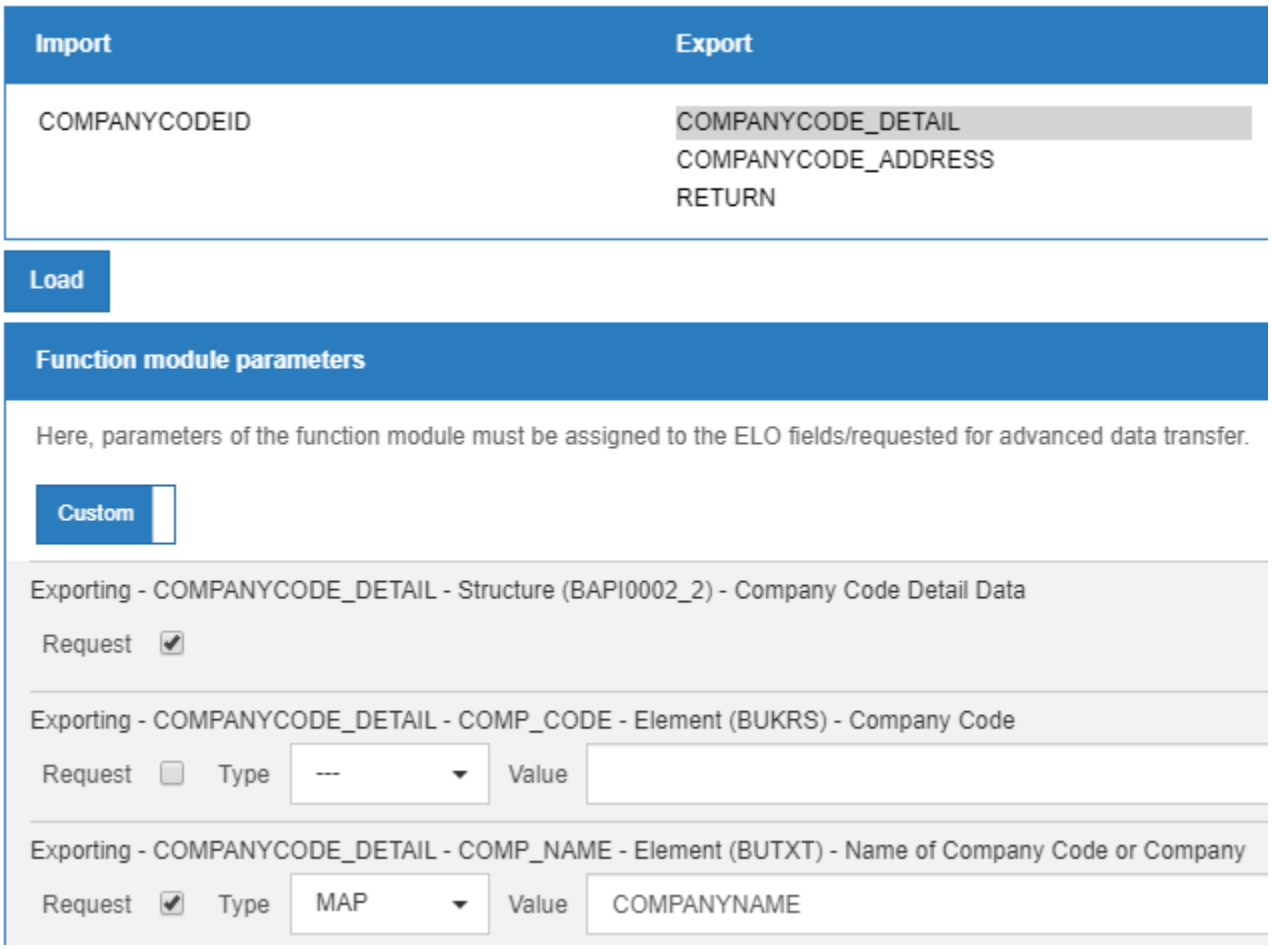

Fig.: 'Configure function module' example – Parameters of the 'COMPANYCODE_DETAIL' export function module

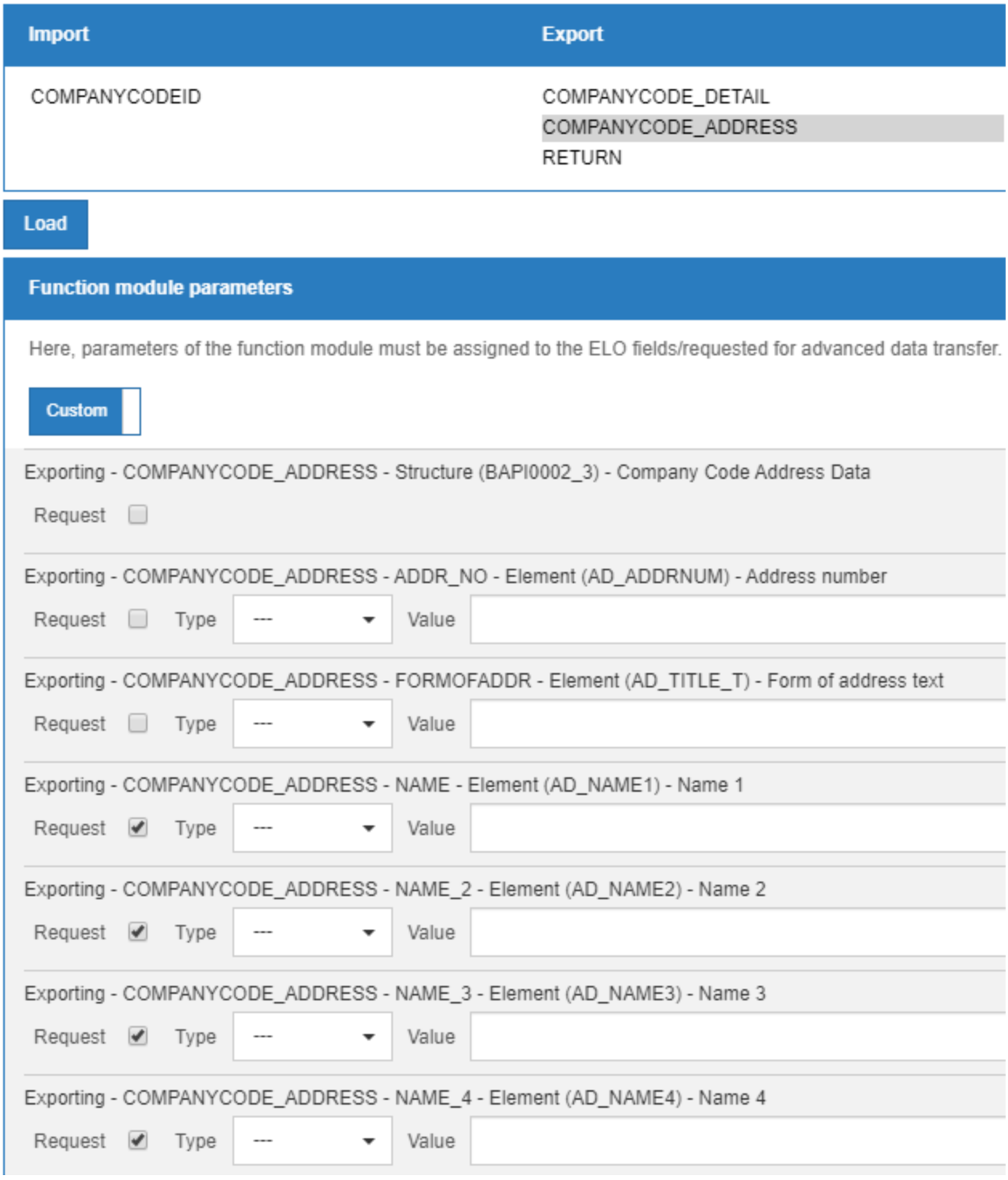

Fig.: 'Configure function module' example – Parameters of the 'COMPANYCODE_ADDRESS' export function module

In this example, a folder is created in the ELO directory using the *Folders* metadata form. By default, this is assigned the field *ELOINDEX*, which is used here for the success or error message when calling the function module.

In the figure *'Configure function module' example – Settings for the 'Advanced data transfer' area*, the line for advanced data transfer contains the following Handlebars string:

```
{{{result.E.COMPANYCODE_ADDRESS.NAME}}}
{{{result.E.COMPANYCODE_ADDRESS.NAME_2}}}
{{{result.E.COMPANYCODE_ADDRESS.NAME_3}}}
{{{result.E.COMPANYCODE_ADDRESS.NAME_4}}}
```

As can be seen in the figure *'Configure function module' example – Parameters of the 'COMPANYCODE_ADDRESS' export function module*, the requested parameters `COMPANYCODE_ADDRESS.NAME`, `COMPANYCODE_ADDRESS.NAME_2` etc. do not need to be mapped to an ELO field in order to use them for advanced data transfer.

To execute the created function module successfully, you need an object in ELO with the field *ELOINDEX* and a map field `COMPANY_CODE` with the value `1000`.

Use in a workflow

The aforementioned function module can be executed in a workflow by creating a distribution node.

This requires the following additional modifications:

- End script: *erp.sap.conn.ix.functions.FmExecute*
- Script content:

```
{
  "configFile": "/erp.sap.conn/Configuration/functionModuleMappings.config",
  "configName": "getCompanyDetail"
}
```

Use in a script

You can execute the function module from a script using an RF call. This works in the context of ELO Indexserver scripting as well as in ELOas.

```
sol.common.IxUtils.execute("RF_erp_sap_conn_function_FmExecute", {
  objId: sord.id,

  configFile: "/erp.sap.conn/Configuration/functionModuleMappings.config",
```

```
  configName: "getCompanyDetail",
})
```

Required permissions

To use the functions of ELO Connectivity Pack – Datatransfer, you need to ensure that the required permissions are assigned in SAP. All SAP users in the *connectivity.config* file require the permissions in the following subchapters, depending on the range of functions used.

Permissions – Search helps

Permissions for executing search helps and querying search help parameters:

Permissions object	ACTVT (Activity)	16 (Execute)
S_RFC	RFC_NAME	SYSU
S_RFC	RFC_NAME	/ELO/CONNECT_GENERIC
S_RFC	RFC_NAME	RFC_METADATA_GET
S_RFC	RFC_TYPE	FUGR

Function module mapping permissions

The rights are always associated with the specific function module. The following table lists the permissions in the SAP system that are required for loading the parameters of SAP function modules into the ELO *functionModuleMapping* configuration form. Normally, you must add at least the name of the function module.

Permission for the ELO *functionModuleMapping.conf* configuration interface:

Permissions object	Field	Value
S_RFC	RFC_NAME	SYST
S_RFC	RFC_NAME	SYSU
S_RFC	RFC_NAME	/ELO/CONNECT_GENERIC
S_RFC	RFC_NAME	/ELO/GET_FM_INTF
S_RFC	RFC_TYPE	FUGR
S_RFC	RFC_TYPE	FUNC
S_RFC	ACTVT (Activity)	16 (Execute)
S_DEVELOP	ACTVT (Activity)	03 (Display)
S_DEVELOP	ACTVT (Activity)	MA (Switch off Mod. Assistant)
S_DEVELOP	OBJTYPE (Object type)	FUGR

Permission for executing *functionModuleMapping.conf*:

Permissions object	Field	Value
S_RFC	RFC_NAME	SYST
S_RFC	RFC_NAME	SYSU
S_RFC	RFC_NAME	/ELO/CONNECT_GENERIC

Permissions object	Field	Value
S_RFC	RFC_TYPE	FUNC
S_RFC	RFC_TYPE	FUGR
S_RFC	ACTVT (Activity)	16 (Execute)
S_CTS_ADMI	CTS_ADMFCT	TABL