

d.velop

d.velop smart invoice:
administration

Table of Contents

1. d.velop smart invoice: administration	3
1.1. Basic information about the application	3
1.1.1. About d.velop smart invoice	3
1.2. Installing and uninstalling (on-premises)	3
1.2.1. System requirements	3
1.2.2. Installing d.velop smart invoice	3
1.2.3. Installing updates for d.velop smart invoice	4
1.2.4. Enabling the default port for d.velop smart invoice	4
1.3. Configuring d.velop smart invoice	4
1.3.1. Configuring steps and connections	4
1.3.2. Interacting with d.velop documents	7
1.3.3. Setting up a simple release with limited complexity	10
1.3.4. Information on complex releases (release step with release matrix)	11
1.3.5. Configuring fields	24
1.3.6. Configuring the layout	25
1.3.7. Setting up formal and factual verification	26
1.3.8. Setting up advanced verification	27
1.3.9. Useful information about the clearing house	28
1.3.10. Configuring buckets	29
1.3.11. Configuring integrations	30
1.3.12. Working with due dates	32
1.3.13. Working with escalations	32
1.3.14. Working with goods receipts	34
1.3.15. Setting up the transfer of line items from d.velop document reader invoice (on-premises)	39
1.3.16. Performing a duplicate check	41
1.3.17. Assigning workflows to repositories	41
1.3.18. Information about conditions	41
1.3.19. Displaying the step comparison view	45
1.3.20. Useful information to know about transferring master data	45
1.3.21. Useful information to know about invoices with reference to a PO	46
1.3.22. Useful information about calculations in d.velop smart invoice	47
1.4. Additional information sources and imprint	51

1. d.velop smart invoice: administration

1.1. Basic information about the application

In this chapter, you will find product hints and general information.

1.1.1. About d.velop smart invoice

This software is used to process your incoming invoices and helps you simplify and optimize invoice processing.

1.2. Installing and uninstalling (on-premises)

In this chapter, you will find information on installing and uninstalling d.velop smart invoice.

1.2.1. System requirements

The following technical requirements must be met to install d.velop smart invoice.

Random access memory

- At least 2 GB of additional RAM is required to operate d.velop smart invoice.

Server operating systems

- Microsoft Windows Server 2012 R2, 64 bit
- Microsoft Windows Server 2016, 64 bit
- Microsoft Windows Server 2019, 64 bit

Databases

- Microsoft SQL Server 2012
- Microsoft SQL Server 2014
- Microsoft SQL Server 2016
- Microsoft SQL Server 2017

Required applications

- d.velop infrastructure components
- d.3one, version 1.8.0 Rising 9 or later
- d.ecs task, version 1.1 or later

Recommended browser

- Google Chrome 78
- Mozilla Firefox 70
- Apple Safari 12
- Microsoft Internet Explorer 11 (this browser is not suitable for administration)

1.2.2. Installing d.velop smart invoice

Suppose you want to install d.velop smart invoice on a server.

This is how it works

1. Download the current setup file.
2. Run the setup file.
3. Start the **si-configure.exe** application in the installation directory as the administrator.

4. Enter the details for the connection to the Microsoft SQL Server database. The user entered must have sufficient permissions to create tables.
5. Enter the processes listed in the **processes.txt** file into d.3 process manager.
6. Start the processes.

1.2.3. Installing updates for d.velop smart invoice

Suppose you want to update an existing installation of d.velop smart invoice to a current version. Make sure that your system meets the current system requirements.

This is how it works

1. Download the latest version of d.velop smart invoice.
2. Quit all d.velop smart invoice processes in d.3 process manager.
3. Run the installation wizard.
4. Start the **si-configure.exe** application in the installation directory as the administrator.
5. Start the d.velop smart invoice processes in d.3 process manager.

1.2.4. Enabling the default port for d.velop smart invoice

The port for d.velop smart invoice is set to port **14084** by default.

1.3. Configuring d.velop smart invoice

In this chapter you will find information about configuration and other settings.

1.3.1. Configuring steps and connections

A workflow consists of steps and connections between these steps. A step is a stage in the workflow, e.g. verification or payment release. Each step is assigned at least one assignee who receives a task within his or her own task list. For each step, it is defined which fields are displayed (for example **gross amount** or **supplier**) as well as their layout.

You create connections between steps to enable the workflow to move from one step to the next, for example from verification to payment release. Each connection has an origin step and a destination step and works only in that direction. You can also link connections to simple conditions such as **amount divided among line items**. In addition, you can trigger actions such as updating a connected system.

Adding a new step

Suppose you want to add the new step **Verification** to an existing workflow.

This is how it works

1. Click **Steps > New step**.
2. Give the step the title **Verification**.
3. If necessary, select a user finder.
4. Use **Step type** to specify the appropriate step type.
5. Click **Add**.
6. Continue with the configuration of the new step.

Specifying the assignee for a step

Each step has one or more assignees. You can use user finders to define which users and groups receive a step for editing. User finders are Urkel expressions and are evaluated for each user and each group.

Suppose you want to assign the **Release** step to the **Management** group.

This is how it works

1. Navigate to the **Release** step.

2. Switch to the **General** perspective.
3. Click next to the **User finder** field and click **New**.
4. Enter a description, e.g. **Approver**, under **User finder**.
5. Enter the value `user.name = "Management"` under **Condition**.
6. Click **Add**.

Sending a step to multiple assignees simultaneously

You can assign a step to multiple assignees at the same time. As soon as one of the assignees completes the step, the task is removed from all assignees' task lists.

Suppose you want to assign the **Accounting verification** step to the **Goods accounting** and **Cost accounting** groups at the same time.

This is how it works

1. Navigate to the **Accounting verification** step and create a new user finder.
2. Give the user finder the title **Reviewer**.
3. Under **Condition**, enter the value `user.name IN ("Goods accounting", "Cost accounting")`.
4. Select the option **Use all hits**.
5. Click **Save**.

Adding a connection between two steps

Suppose you want to add a connection between two steps. Connections are always located downstream of an action, which is displayed to the users as an option during editing. You can configure an action so that the workflow will automatically choose between multiple connections. For example, depending on the total amount of the invoice, it may be possible for the user to skip the **Payment release** step when they click on an action.

Which connection is selected is determined by the first criterion met based on the top-down principle.

Suppose you want to add a connection from the **Verification** step to the **Payment release** step.

This is how it works

1. Navigate to the **Verification** step.
2. Switch to the **Actions** perspective.
3. Select an existing action or create a new one, for example **Next**.
4. Click **New connection** within the action.
5. Enter a name for the new connection, for example **To payment release**. This name is displayed when users hover the mouse over the action.
6. From the **To** list, select the **Payment release** step.
7. Use the condition to specify when the connection can be selected. The simplest possible condition is **true**.
8. Click **Add**.

Specifying a condition for a connection

Suppose you want a connection to be selectable only under certain circumstances. You can accomplish this by specifying a condition for the connection. Conditions are described in the Urkel expression language.

Suppose you want to allow a connection to be selected only if the amount of the invoice has been completely divided among the line items. The existing function `voucher.amount_allocated_tolerance?` can be used here. This function checks if the gross amounts of the header data and line item data match. You can specify a tolerance between the net amounts, such as ten cents, as described in the following example:

This is how it works

1. Navigate to the step you want to link from.
2. Switch to the **Actions** perspective.
3. Open the editing screen for the connection.
4. Click the menu with the three horizontal dots next to the connection and select **Edit**.
5. In the **Condition** field, enter the value `voucher.amount_allocated_tolerance?(0, 10)`.
6. Click **Update**.

Setting a workflow start

Workflows always start from a special start connection. This connection offers the same options as connections between two steps.

Suppose you want the workflow to always start at the **Distribution** step.

This is how it works

1. Navigate to the workflow settings.
2. Switch to the **Workflow start** perspective.
3. Select **New connection** to create a new start connection.
4. Enter an appropriate title, e.g. **To distribution**.
5. From the **To** list, select the **Distribution** step.
6. Enter the value `true` under **Condition**.
7. Click **Add**.

Specifying workflow start conditions for documents

Your workflow definition is linked to a repository and automatically initiates a workflow when a document is saved to a particular category in the repository. You can specify the conditions under which a workflow will be initiated. If there are multiple workflow definitions, you can also specify which workflow will be started. You can make these assignments using Urkel expressions.

Suppose you have linked the workflow **Cost invoices** to repository A, and the workflow **Goods invoices** to repository B. You want the **Cost invoices** workflow to start for every new document. However, the workflow **Goods invoices** should start only for the documents that have client number 23.

This is how it works

1. Navigate to the settings for the **Cost invoices** workflow.
2. Switch to the **General** perspective.
3. Enter the value `true` under **Start connections**.
4. Save the configuration.
5. Navigate to the settings for the **Goods invoices** workflow.
6. Switch to the **General** perspective.
7. Enter the value `voucher.company.nr = "23"` under **Start connections**.

Offering users the option to select the assignee

Suppose you want users to be able to choose the assignee for the **Release** step from among the groups **Administration**, **Sales**, and **Development**.

This is how it works

1. Navigate to the **Release** step and create a new user finder.
2. Give the user finder the title **Approver**.
3. Under **Condition**, enter the value `"Administration" IN user.group_names OR "Sales" IN user.group_names OR "Development" IN user.group_names`.

4. Select the **Offer choice** check box.
5. Click **Save**.

1.3.2. Interacting with d.velop documents

d.velop smart invoice uses d.velop documents to store invoices, invoice attachments and workflow logs, among other things. d.velop documents is accessed via a source system, which is independent of the repository's specific category and property structure. There are only a few categories and properties that must be present and assigned for d.velop smart invoice to function.

Category: Invoices

- Company number
- Company name
- Workflow participants
- Created on (system property)

Category: Workflow logs

- Doc ID workflow

You can use the **Workflow document ID** property in the **Workflow logs** category to establish the connection between the invoice and the log.

Beyond this minimal set of fields, you are free to choose the repository structure.

In order for d.velop smart invoice to connect to a repository, you need to enter an API key. The user associated with the API key must have read and write permissions in all repositories for the categories used. In particular, you must grant the user the permission **Change attribute in release**.

Deactivating the synchronization of guest users in SharePoint

You want to deactivate the synchronization of external users such as guests (e-mail addresses with #EXT#) in SharePoint. This then stops newly created guest users from being synchronized and sets existing guest users as inactive.

This is how it works

1. In the d.velop smart invoice administration area, go to **User/Groups**.
2. Under **Interval (Minutes)**, activate the relevant checkbox to stop the synchronization of guest users.
3. Save the synchronization settings. The changes are applied during the next synchronization process.

Setting up assignments

To use a repository with d.velop smart invoice, you need to assign the sources **smart invoice - invoice** and **smart invoice - protocol** to the corresponding categories and properties in your d.3 repository.

Suppose that your d.3 repository contains the categories **Incoming invoices** and **Workflow logs**.

This is how it works

1. Click the **Mappings** tile.
2. Create a new assignment.
3. Select **smart invoice - invoice** as the source.
4. Select **Incoming invoices** as the category.
5. Assign the d.velop smart invoice property **d.3ecm - creation date** to the system property **Created on**.
6. Assign the required d.velop smart invoice properties to the category properties. You must assign at least the properties **Company number**, **Workflow participants** and **Company name**.

7. Save the assignment.
8. Create a new assignment.
9. Select **smart invoice - protocol** as the source.
10. Select **Workflow logs** as the category.
11. Assign the required d.velop smart invoice properties to the category properties. You must assign at least the **Workflow doc ID** property.
12. Save the assignment.
13. Navigate to the d.velop smart invoice administration interface.
14. Select **Repositories** to switch to the overview of d.3 repositories. If the repository and assignments have been set up correctly, a check mark appears in the **Status** column.

The d.velop documents repository is now ready for use.

Granting permissions for invoice documents

All users need repository read permissions for the invoices they process in d.velop smart invoice. The multi-value property **Workflow participants** is used for this purpose. d.velop smart invoice populates this field with a list of all users and groups to whom the invoice is assigned throughout the workflow. In the repository, you need to set up an appropriate permission profile that grants these users and groups read access for the invoice documents.

Setting up a permission profile for d.ecs identity provider support (on-premises)

If you are using d.velop documents with support for d.ecs identity provider enabled, you can use the **@D3IDPUSER** and **@D3IDPGROUP** macros to grant read permissions for the invoice documents to the workflow participants. Unless there are reasons not to, use d.ecs identity provider support instead of LDAP support.

Suppose you are using a document type with the name **Incoming invoices**.

Note

When setting up the mappings, make sure that you have mapped the **Workflow participants IDs (workflowParticipantsIds)** (source) field to a document property.

This is how it works

1. Open the d.velop documents administration and switch to editing mode.
2. Open **d.3 restriction set manager**.
3. Create a new restriction set named **Workflow participant IDs** to restrict documents by users and groups.
4. Create a new set assignment of the type **No mapping**.
5. Enter the values **@D3IDPUSER** and **@D3IDPGROUP** as filters in two separate lines.
6. Save the set assignment.
7. Create a new document class for the document type **Incoming invoices**.
8. Enter the value **@D3SET(Workflow participants IDs)@D3SET(Workflowbeteiligte IDs)** in the field assigned to the d.velop smart invoice field **Workflow participant IDs (workflowParticipantsIds)Workflow participant IDs (workflowParticipantsIds)**.
9. Name the document class **Workflow participants incoming invoices**.
10. Save the document class.
11. Create a new permission profile named **smart invoice user**.
12. Add the document class **Workflow participants incoming invoices**.
13. Grant read permissions for this document class.
14. Save the permission profile.

Setting up a permission profile for LDAP support (on-premises)

If you use d.velop documents with support for LDAP enabled, you can use a combination of the macros **@D3GROUP** and **@D3USER_OPTFIELD_NN** to grant the workflow participants read permissions for the invoice documents.

You can use the **@D3GROUP** macro to grant users permissions to invoices that were assigned to a group to which the user belongs during the workflow. Please note that d.velop smart invoice uses the groups from d.ecs identity provider, which are usually groups from connected external user management systems, such as Microsoft Active Directory. For the permissions to be correctly assigned via **@D3GROUP**, each user group that is relevant for d.velop smart invoice must have exactly the same name in d.velop documents. Additionally, you must use the d.velop documents LDAP support function to ensure that the users belonging to a group in the connected LDAP system are also assigned to the corresponding group in d.velop documents. Details can be found in the administration manual for d.velop documents.

You can use the **@D3USER_OPTFIELD_NN** macro to grant users permissions to invoices that were assigned to the users directly during the workflow and were therefore not assigned via a group. To do this, enter the corresponding login name of the d.velop documents user in an optional field in the LDAP support function. Note that the optional field must match the login name used by d.ecs identity provider. If you are using Microsoft Active Directory, you can use the **sAMAccountName** field.

Suppose you are using a document type named **Incoming invoices** and you have copied the user's login name to Optional field 1. For the document type, there is a multi-value property named **Workflow participants** to which you have assigned the d.velop smart invoice field **Workflow participants**.

This is how it works

1. Open d.3 admin and switch to edit mode.
2. Open **d.3 restriction set manager**.
3. Create a new restriction set named **Workflow participants** to restrict documents by users and groups.
4. Create a new set assignment of the type **No mapping**.
5. Enter the values **@D3USER_OPTFIELD_01** and **@D3GROUP** as filters in two separate lines.
6. Save the set assignment.
7. Create a new document class for the document type **Incoming invoices**.
8. Enter the value **@D3SET(Workflow participants)** in the **Workflow participants** field.
9. Name the document class **Workflow participants incoming invoices**.
10. Save the document class.
11. Create a new permission profile named **smart invoice user**.
12. Add the document class **Workflow participants incoming invoices** to the new permission profile.
13. Grant read permissions for the document class.
14. Save the permission profile.
15. Use the LDAP support function to assign the permission profile to all users participating in the workflow.

Defining a delegate during absences

If you report yourself as absent in d.velop documents and define a delegate, your delegate can access your task list.

Your delegate can only view tasks that have been personally addressed to you. Your delegate cannot view any tasks that were sent to you as part of a group membership. As soon as your delegate opens a task, d.velop smart invoice adds the delegate to the workflow participants.

To ensure that your delegate can also see the document attachments for the tasks, your delegate must have an appropriate permission profile in d.velop documents.

Note

In d.velop documents, permission problems may occur in rare cases due to the configuration of the permission profiles. That means the delegate is not immediately authorized to view the attached invoice document when opening a task.

This can happen when permission profiles are used to grant the permissions for invoice documents based on the list of workflow participants. In such cases, the invoice document is loaded more quickly than the delegate can be added to the list of workflow participants. Closing the task and opening it again rectifies the problem with the permission.

1.3.3. Setting up a simple release with limited complexity

Complex release rules are not always necessary. In many cases, invoices are only released by one user group, such as management. Sometimes, separate release of an invoice is only required as of a specific invoice amount. In this case, you can configure one or, if necessary, multiple standard workflow steps to map this release.

Setting up a one-stage release

Suppose that the company management is to be responsible for releasing all documents. You have created the **Management** user group for this purpose. The last step before release is **Verification**. After the release comes the **Accounting** step, to allow for a final check before posting to the ERP system. You have already set up the **Verification** and **Accounting** steps.

This is how it works

1. Create a new step with the title "Release".
2. Create a user finder that selects the **Management** user group.
3. Switch to the settings for the **Verification** step.
4. Switch to the **Actions** perspective.
5. Create an action with the title **Next**, if not already created.
6. Within the **Next** action, create a connection to the **Release** step with the following condition: **voucher.verified?**
7. Switch to the settings for the **Release** step.
8. Switch to the **Actions** perspective.
9. Create a new action with the title **Release**.
10. Within the **Release** action, create a connection to the **Accounting** step with the following condition: **true**.
11. In the connection settings, under **Release**, select the option **Approve**.

You have created a one-stage release that must always take place.

Setting up a two-stage release dependent on the invoice amount

Suppose that the **Head of purchasing** user group is to be responsible for releasing all documents. For documents with a net value of more than EUR 5,000.00, an additional release by the **Management** user group is to take place. You have already set up the **Purchasing release** step for the first release by the **Head of purchasing** user group. This step contains the action **Release** with a connection to the **Accounting** step.

This is how it works

1. Create a new step with the title "Management release".
2. Create a new user finder that selects the **Management** user group.
3. Create suitable actions and step connections, e.g. a transfer to the **Accounting** step.

4. Switch to the **Purchasing release** step.
5. Switch to the **Actions** perspective.
6. Within the **Release** action, create a new connection to the **Management release** step with the following condition: `voucher.net_amount >= 500000`.
7. In the settings for the connection, under **Release**, select the option **Approve**.
8. Under **Release**, select the option **Approve**.
9. Drag and drop the new connection over the existing connection to the **Accounting** step.

You have set up a two-stage release dependent on the invoice amount. If you now click on the **Release** action in the **Purchasing release** step, the software automatically selects the connection to the **Management release** step if the document has a net amount of at least EUR 5,000.00. Otherwise, it selects the connection to the **Accounting** step.

1.3.4. Information on complex releases (release step with release matrix)

In some companies, release rules can be very complex. To map extensive and complex release rules, workflow steps of the type **Release** exist in d.velop smart invoice. Release steps have a release strategy and use a release matrix for determining the appropriate people for releasing the invoice.

If the release in the verification and release workflow is to be based on the document data, particularly the data in the line items of a document, it makes sense to use a release step with release matrix.

Background knowledge on release matrices

This section provides detailed information on how a release matrix works and how you to set up a release matrix. You can find a practical example of a release based on cost centers here: [Setting up a release matrix and a release step](#).

With a release matrix, you can automatically determine the appropriate people for processing an invoice based on the invoice data. Assignees can be determined, for example, based on the cost center or the cost units. The relevant data used to automatically determine the assignee depends on the structure of the release matrix. Each release matrix consists of the two fixed columns **Assignee** and **Release limit**. You can use up to twenty additional columns from the fields of an invoice, e.g. **Client**, **Cost center** or **Cost unit**. The rows of the release matrix then determine which people are possible assignees for which values.

Example of a release matrix

Person	Release limit	Client	Cost center	Cost unit
Susanne Meyer	EUR 10,000.00	docures Deutschland AG	2000	P001
Theresa Taylor	EUR 20,000.00	docures Deutschland AG	5000	P001

In this example, the employee Susanne Meyer is allowed to release amounts up to EUR 10,000.00 for the client docures Deutschland AG as well as for the cost center 2000 and the cost unit P001. The first row of the release matrix contains the corresponding values. Employee Theresa Taylor is authorized to release up to EUR 20,000.00 for the same client and cost unit, but only for cost center 5000. The second row of the release matrix contains the corresponding values.

Note

The evaluation of the release matrix is case-sensitive.

Example: The comparison value P001 in the "Cost unit" column of a release matrix matches only the value P001 of the "Cost unit" field in a workflow line item. If the value for the "Cost unit" field in a workflow line item is p001, the matrix row is not taken into account.

Cost objects and release limits

The configurable fields of a release matrix that determine for which invoices an entry in the release matrix is suitable are collectively called the cost object for an entry. A cost object can consist of client, cost center and cost unit. The release limit for each entry in the matrix refers to the associated cost object, not to the entire invoice. Even if the total amount of the invoice is higher than the release limit set for one person, this person is allowed to release their part of the invoice. The condition for releasing partial amounts is that the relevant portion of the invoice is within the release limit.

Example of release limits for cost objects

Suppose an invoice has a total amount of EUR 25,000.00. The invoice contains partial costs of EUR 17,000.00 for cost center 5000 and cost unit P001 as well as partial costs of EUR 8,000.00 for cost center 2000 with the same cost unit.

The following configuration is allowed, as the individual release limits of both persons are met. Theresa Taylor and Susanne Meyer can release their respective partial amounts, even though the total amount of the invoice exceeds their release limits.

Cost center	Cost unit	Amount	Person
2000	P001	EUR 8,000.00	Susanne Meyer
5000	P001	EUR 17,000.00	Theresa Taylor

The following configuration is not allowed, because Theresa Taylor's release limit is exceeded.

Cost center	Cost unit	Amount	Person
2000	P001	EUR 12,000.00	Susanne Meyer
5000	P001	EUR 13,000.00	Theresa Taylor

Placeholders in matrix fields

You can use placeholders instead of exact values in the fields of a matrix. The following placeholders are allowed:

- An asterisk (*) after a value means that the matrix row is valid for all invoices or line items that begin with the specified value before the asterisk in the corresponding field.
- An empty field in a matrix row means that all values are permitted in the corresponding field for the invoice or line item.
- A dollar sign (\$) means that the invoice or line item must not have a value in the corresponding field.

Example of placeholders

Person	Release limit	Cost center
A	EUR 5,000.00	20*
B	EUR 6,000.00	\$
C	EUR 7,000.00	

In this case, person A would be allowed to release all cost centers up to EUR 5,000.00 that start with 20, for example 2050 or 2031. Person B would only be allowed to release invoices when no cost center is selected. Person C can release invoices for all cost centers.

How persons are selected for processing

The matrix determines persons for processing invoices or parts thereof. The most suitable person is selected based on the matrix. The invoice lines are first summed by cost object in order to determine the total amount per cost object. The matrix rows are then used to determine the person who best matches the cost object and whose release limit is higher than the invoice amount for the cost object. A matrix row is most suitable if all fields match between the cost object and matrix row and the release limit is not exceeded.

The best matrix row is determined according to the following principle:

1. For each field, a score is calculated that reflects how well the contents of the matrix row field match the contents of the cost object field on the invoice. Information on how the score is determined is provided in the following section.
2. The score for each field is weighted according to the order of the columns in the matrix. The front fields get the lowest weight, while the back fields get the highest.
3. The row with the highest score and a sufficient release limit is selected. If multiple rows have the same score, the first suitable row is selected.

The score for each field is determined according to the following principle:

1. The highest score is assigned when the field contents of the matrix and the cost object match exactly.
2. The second highest score is assigned if the cost object field is empty and the matrix row contains the placeholder \$.
3. The third highest score is assigned if the field in the matrix row ends with the placeholder * and the value before the placeholder matches the contents of the cost object field (prefix match). The score in this case is higher when the prefix is longer and lower when the prefix is shorter.
4. The lowest score is assigned if the field in the matrix row is empty.
5. In some cases, no score will be awarded. In this case, the entire row is ineligible and will not be taken into account.

Example of selecting persons for processing

Person	Release limit	Client	Cost center	Cost unit
Susanne Meyer	EUR 5,000.00	docures Deutschland AG	2000	P001
Theresa Taylor	EUR 7,000.00	docures Deutschland AG	2000	
Peter Gibbons	EUR 2,000.00	docures Deutschland AG	2000	\$
Maria Management	EUR 50,000.00	docures Deutschland AG		

For the cost object “docures AG, 2000, P001”, Susanne Meyer, Theresa Taylor and Maria Management match. Susanne Meyer is preferred if the amount is below EUR 5,000.00. Between EUR 5,000.00 and EUR 7,000.00, Theresa Taylor is responsible. If the amount is greater than EUR 7,000.00, the invoice goes to Maria Management. Peter Gibbons is eliminated because he is not approved for any cost unit (placeholder \$).

For the cost object “docures AG, 2000, with no cost unit”, Peter Gibbons, Theresa Taylor and Maria Management match. Peter Gibbons is preferred if the amount is below EUR 2,000.00. Between EUR 2,000.00 and EUR 7,000.00, Theresa Taylor is responsible. All invoices above this amount must be cleared by Maria Management. Susanne Meyer is eliminated because the cost unit P001 is not specified for the cost object.

Note

Note that factual reviewers are not considered if you have enabled the dual control principle.

Setting up a release matrix and a release step

You can set up a release step that reacts dynamically to the cost centers entered for the line items and determines the person responsible for release on this basis. The following steps are required:

1. Create a new release matrix. You use the release matrix to define who is authorized to release which cost center and up to which limit.
2. Create a new step for the release.

3. Link the step to the release matrix.
4. Integrate the new step into the workflow.

The following subsections describe the above steps using the following example scenario:

Suppose you have already set up the **Verification** and **Accounting** steps and would now like to configure a workflow step of the type **Release** in d.velop smart invoice. You want the workflow to proceed to the **Release** step after the **Verification** step and continue to the **Accounting** step once the release has been granted. If the release is denied, the workflow is to return to the **Verification** step. The release step determines the right people for release based on the cost centers and the line items of a workflow.

Creating a release matrix

The following action steps refer to the example scenario under [Setting up a release matrix and a release step](#).

A release matrix dynamically determines people for release based on the entered invoice information. The first step is to create a new matrix and add the **Cost center** column as a decision criterion.

Creating a release matrix – this is how it works

1. In the d.velop smart invoice administration interface, navigate to **Release matrices**.
2. Click **New matrix**.
3. Enter the name "Release by cost center" and save the matrix.
4. Select **Cost center** and click **Add column**.
5. Click **Save**.

You have configured the structure of the new release matrix and can now add matrix rows. With the individual matrix rows, you specify which person is authorized to release which cost center and up to which invoice amount.

Adding a matrix row – this is how it works

1. In the d.velop smart invoice administration interface, navigate to **Release matrices**.
2. Click the matrix that you created earlier.
3. Click **Add row**.
4. Select the relevant person.
5. In the **Limit** field, enter the invoice amount up to which the person is authorized to release.
6. Enter the cost center for which the person is responsible in the **Cost center** field.
7. Save the new row.

Repeat this process for additional matrix rows if necessary. You can now add a release step to the workflow and connect this release step to the release matrix.

Creating a release step

The following action steps refer to the example scenario under [Setting up a release matrix and a release step](#).

You have already set up a release matrix and now want to create a release step.

Creating a new step – this is how it works

1. Navigate to the administration interface for the workflow.
2. Under **Steps**, click **New step**.
3. Enter the title "Release".
4. Select **Release** as the step type.
5. In the **Assignee** area, under **Matrix**, select the matrix you created previously.

6. Under **Strategy**, select the option **Direct**.
7. Under **Net amount or gross amount**, select whether the release limits entered in the release matrix should be applied to the net or gross amount of the created cost objects. You can ignore the other configuration options for now.
8. Click **Add**.

The actions in a workflow step of the type **Release** have a special significance. In the context of the release, a type must be specified for the actions. A workflow step of the type **Release** must have at least one action of the type **Confirm** and one action of the type **Reject**.

Configuring actions – this is how it works

1. Switch to the **Actions** perspective.
2. Create a new action and enter the title "Release".
3. Select **Confirm** as the type. You can ignore the configuration option under **Default action for this step** for now.
4. Click **Add**.
5. In the new action **Release**, create a new connection and enter the name "To accounting".
6. Select the **Accounting** step as the destination for the connection.
7. Enter the value **true** under **Condition**.
8. Create another action and enter the title "Reject".
9. Select **Reject** as the type.
10. In the new action **Reject**, create a new connection and enter the name "To verification".
11. Select the **Verification** step as the destination for the connection.
12. Enter the value **true** under **Condition**.

You have created the workflow step of the type **Release** and can now configure the incoming step connection.

Note

In the step connection, under **Release**, select the option **No effect**.

Creating the step connection for the release step

The following action steps refer to the example scenario under [Setting up a release matrix and a release step](#).

You have already configured the release matrix and the release step. You have connected the release matrix to the release step. You have also set up the release step connections for the cases **Release** and **Reject**. You now require an incoming step connection for the release step. Suppose that the **Verification** step already has the **Next** action with a step connection that is now to be replaced with the new step connection to the release step.

This is how it works

1. Navigate to the **Verification** step in the d.velop smart invoice administration interface.
2. Switch to the **Actions** perspective.
3. In the **Next** action, click the three dots next to the step connection that you want to replace.
4. Click **Delete**.
5. In the **Next** action, click **New connection**.
6. Enter the name "To release".
7. Select the **Release** step as the destination for the connection.
8. Enter the value **voucher.verified?** under **Condition**.

You have now successfully added the new step **Release** to the workflow and connected it to the release matrix. When an invoice is sent from verification to release, d.velop smart invoice automatically determines the correct persons for release based on the cost centers and amounts entered.

How the "bottom to approver" release strategy works and how to set it up

By default, a document with the **direct** release strategy is delivered to the most suitable person with the lowest release limit above the invoice amount.

Example 1

There is an invoice for EUR 12,000.00 for cost center 2000.

Person	Release limit in EUR	Cost center
A	20.000,00	2*
B	15.000,00	20*
C	10.000,00	200*

In this case, only person B receives the document for release. While person C has the highest specificity, the corresponding release limit is insufficient. Person B is the best match for the cost object and has a sufficient release limit.

Example 2

There is an invoice for EUR 12,000.00 for cost center 2000.

Person	Release limit in EUR	Cost center
A	20.000,00	2000
B	15.000,00	20*
C	10.000,00	200*

In this case, only person B receives the document for release. Person A has the highest specificity and is therefore the most suitable person for the release.

The **bottom to approver** strategy allows you to take a different approach than the **direct** strategy. With **bottom to approver**, the document passes through several release levels, starting with the matrix row that has the highest degree of specification, regardless of the release limit. The system then passes through all specification levels, starting from the highest degree of specification. For each specification level in which no person is authorized to release based on the amount limit, the person with the highest release limit receives the document. If a specification level includes one or more people who are authorized to release based on the amount limit, the person with the lowest release limit that still exceeds the amount to be released receives the document. The process ends when the first person authorized to release based on the amount limit receives the document for release.

Example 3

Person	Release limit in EUR	Cost center
A	1.000,00	2000
B	2.000,00	200*
C	3.000,00	20*
D	4.000,00	2*
E	10.000,00	*

There is an invoice for EUR 2,500.00 for cost center 2000.

- This invoice would first be sent to person A, then person B, and then person C for release.

- The user finder starts at the matrix row with the highest degree of specification (person A) and proceeds through all specification levels until it gets to the first person authorized to release the document (person C).

There is an invoice for EUR 3,500.00 for cost center 2000.

- This invoice would first be sent to person A, then person B, then person C, and then person D for release.
- The user finder starts at the matrix row with the highest degree of specification (person A) and proceeds through all specification levels until it gets to the first person authorized to release the document (person D).

There is an invoice for EUR 4,500.00 for cost center 2000.

- This invoice would first be sent to person A, then person B, then person C, then person D, and then person E for release.
- The user finder starts at the matrix row with the highest degree of specification (person A) and proceeds through all specification levels until it gets to the first person authorized to release the document (person E).

Example 4

Person	Release limit in EUR	Cost center
A	1.000,00	2000
B	2.000,00	2000
C	3.000,00	2000
D	4.000,00	2*
E	10.000,00	*

There is an invoice for EUR 1,500.00 for cost center 2000.

- Only person B would receive this invoice for release.
- Persons A, B and C have the same degree of specification. Person C has the lowest release limit that still exceeds the amount to be released and is also the first person authorized to release the document.

There is an invoice for EUR 2,500.00 for cost center 2000.

- Only person C would receive this invoice for release.
- Persons A, B and C have the same degree of specification. Person C has the highest release limit and is also the first person authorized to release the document.

There is an invoice for EUR 3,500.00 for cost center 2000.

- This invoice would first be given to person C and then to person D for release.
- Persons A, B and C have the same degree of specification. Person C has the highest release limit. Person D is the first person authorized to release the document.

There is an invoice for EUR 4,500.00 for cost center 2000.

- This invoice would first be sent to person C, then person D, and then person E for release.
- Persons A, B and C have the same degree of specification. Person C has the highest release limit. Person E is the first person authorized to release the document.

Setting up a release step according to the "bottom to approver" strategy – this is how it works

1. Navigate to the relevant workflow definition in the d.velop smart invoice administration interface.

2. Select the relevant release step under **Steps**.
3. Navigate to **General > Strategy**.
4. Select the release strategy **Bottom to approver**.

For the **bottom to the approver** strategy to work, the release matrix must map a hierarchy of people. Mapping the hierarchy is simple if, for example, the customer's cost center structure maps a hierarchy. Example: The first two characters of the cost center contain the division, the next two characters contain the department, the next two characters represent the team, etc. In this case, d.velop smart invoice can work through the hierarchy levels from the highest specificity (team) to the lowest specificity (division), thus determining the people for the release. If no hierarchically structured data exists, you can still use the **bottom to approver** strategy by representing the hierarchy using the **Rank** field.

Using the "bottom to approver" strategy with the aid of the "Rank" field – this is how it works

1. Add the **Rank** column to the relevant release matrix.
2. Enter a number between 0 and 99 for each person in the **Rank** column to represent the hierarchy.

Example 5

Person	Release limit in EUR	Rank	Cost center
A	1.000,00	0	2000
B	2.000,00	10	2000
C	3.000,00	20	2000
D	4.000,00	30	2000
E	10.000,00	99	2000

There is an invoice for EUR 2,500.00 for cost center 2000.

- This invoice would first be sent to person A, then person B, and then person C for release.
- Persons A, B and C have the same degree of specification and differ only in rank. Person A has the lowest rank, which, in accordance with the **bottom to approver** strategy, results in this matrix entry having the highest specificity for otherwise equally specific matrix entries. The **bottom to approver** strategy determines the persons from bottom to top, i.e. from the highest specificity in decreasing order of specificity.
- Person C is the first person authorized to release the document.

There is an invoice for EUR 4,500.00 for cost center 2000.

- This invoice would first be sent to person A, then person B, then person C, then person D, and then person E for release.
- The same explanation applies as for the previous scenario.

Note

If a person is included in multiple matrix rows, and the matrix rows match a created cost object more or less well depending on their specificity, only one matrix row is retained for that person. Regardless of the specificity of the matrix rows, only the matrix row with the smallest release limit that still exceeds the amount to be released is retained. If such a matrix row does not exist, i.e. all matrix rows have a release limit below the amount to be released, only the matrix row with the highest release limit is retained. This mechanism can lead to unexpected release sequences, but does not constitute an error.

Example 6

Person	Release limit in EUR	Cost center
A	2.000,00	2000
B	3.000,00	20*
C	4.000,00	2*
B	10.000,00	2000

There is an invoice for EUR 2,500.00 for cost center 2000.

- This invoice would first be sent to person A and then person B for release.
- Person A and B share the same and also the highest specificity. Although person B has a sufficient release limit, they do not receive the document as the first and only person since person B is also included in another matrix row. The other matrix row also matches the cost object, but not quite as well due to the lower specificity. However, since the release limit is the lowest above the amount to be released, the matrix row is retained for determining the release sequence. Because the last matrix row of the table was sorted out, the release sequence is as described.

Sending the document to multiple recipients (ad-hoc groups)

Note

The following section describes a preview feature. If you would like to test this preview feature, contact d.velop support and make reference to this documentation.

Without further configuration, a document is always sent to only one person for release. Therefore, if multiple entries in a release matrix have the same specificity and the same release limit, meaning that all people are equally eligible to release a cost object, only the first suitable person will be selected.

You can configure that entries in a release matrix with the same specificity and release limit are grouped into an ad-hoc group. A document will then always be sent to all matching people for release. The basic mechanisms for determining matching entries in a release matrix remain intact. Note that factual reviewers are not considered if you have enabled the dual control principle.

Example

There is an invoice for EUR 4,000.00 for cost center 2000.

Person	Release limit in EUR	Cost center
A	5.000,00	2000
B	5.000,00	20*
C	5.000,00	2000
D	6.000,00	2000

Person A and person C would receive the invoice for release, as they are the first persons authorized to release according to the matrix. Person B is eliminated because the cost center is less specific than those of persons A and C. Person D is eliminated because, although the release limit is above the amount to be released, it is not the lowest release limit above this amount.

Forcing a dual control principle during release

Note

The following section describes a preview feature. If you would like to test this preview feature, contact d.velop support and make reference to this documentation.

In d.velop smart invoice, you can specify that a second release is always required for amounts exceeding EUR 10,000.00. This forces a second release, even if the release strategy and release matrix employed would require only one release by one person.

How the dual control principle works for the "direct" release strategy

If the amount of a cost object of an invoice exceeds EUR 10,000.00, a separate release by two different people is required.

The release sequence is determined according to the following steps:

1. In the first step, the most suitable person for the cost object with a release limit above the amount to be released is selected. For more information, see [How persons are selected for processing](#).
2. In the second step, the person is selected from the matrix who matches the cost object of the document based on their specificity and who has the highest release limit below the amount to be released. This person is inserted in the release sequence before the person identified in step 1. Step 3 is no longer required because two people have been identified for the release. If no person can be determined in step 2, step 3 follows.
3. In the third step, the person is selected from the matrix who matches the cost object of the document based on their specificity and who has the lowest release limit above the amount to be released. This person is inserted in the release sequence after the person identified in step 1. If no person can be identified in step 3, the requirements for releasing the cost object cannot be met. In this case, d.velop smart invoice displays a corresponding error message.

The release sequence is thus as follows: The person determined in step 2 releases before the person identified in step 1. If no person could be determined in step 2, the person identified in step 1 releases, followed by the person identified in step 3.

Example 1

There is an invoice for EUR 12,000.00 for cost center 2000. For amounts of EUR 10,000.00 and up, two persons must release the document.

Person	Release limit in EUR	Cost center
A	5.000,00	2000
B	20.000,00	2000
C	4.000,00	2000
D	50.000,00	2000

This invoice is sent to person A and then to person B for release.

- Person B is the person who best matches the cost object and whose release limit is the lowest above the amount to be released.
- Persons A, C and D match the document in terms of their specificity. However, person A has the highest release limit under the invoice amount and is therefore included in the release sequence.

Example 2

There is an invoice for EUR 12,000.00 for cost center 2000. For amounts of EUR 10,000.00 and up, two persons must release the document. Person B has already performed the verification. The dual control principle is enabled.

Person	Release limit in EUR	Cost center
A	20.000,00	2000
B	5.000,00	2000
C	30.000,00	2000
D	50.000,00	2000

This invoice is sent to person A and then to person C for release.

- Person A is the person who best matches the cost object and whose release limit is the lowest above the amount to be released.

- Person B matches the document in terms of their specificity. Person B has the highest release limit below the invoice amount. However, person B has already factually verified the invoice and is therefore eliminated for release.
- Since there are no other persons whose release limits are below the amount to be released, persons whose release limits are above the amount to be released are now identified. Persons C and D match the document in terms of their specificity. Person C has the lowest release limit above the amount to be released and is therefore included in the release sequence.

Example 3

There is an invoice for EUR 12,000.00 for cost center 2000. For amounts of EUR 10,000.00 and up, two persons must release the document.

Person	Release limit in EUR	Cost center
A	20.000,00	2000
B	15.000,00	[de]
C	30.000,00	2000
D	50.000,00	2000

This invoice is sent to person A and then to person B for release.

- Person A is the person who best matches the cost object and whose release limit is the lowest above the amount to be released.
- Since there are no persons whose release limits are below the amount to be released, persons whose release limits are above the amount to be released are identified. Persons B, C and D match the document in terms of their specificity. Person B has the lowest release limit above the amount to be released and is therefore included in the release sequence.

How the dual control principle works for the "bottom to approver" release strategy

The **bottom to approver** release strategy is already designed to involve multiple people in the release of a cost object. However, depending on the document data and release matrix, it is still possible that the release of a single person is sufficient. If, in this case, you still want to force a second release for amounts of EUR 10,000.00 or more, activate the dual control principle.

The release sequence is determined according to the following steps:

1. In the first step, the release sequence for the cost object is determined. Further information: [How the "bottom to approver" release strategy works and how to set it up](#). If two or more people are already involved in the release in this step, the dual control principle is already ensured, meaning that steps 2 and 3 are no longer necessary.
2. In the second step, the person is selected from the matrix who matches the cost object of the document in terms of their specificity and who has the highest release limit below the amount to be released. The identified person is inserted at the start of the release sequence. Step 3 is no longer required because two people have been identified for the release. If no person can be determined in this step, step 3 follows.
3. In the third step, the person is selected from the matrix who matches the cost object of the document in terms of their specificity and who has the lowest release limit above the amount to be released. The identified person is inserted at the end of the release sequence. If no person can be identified, the requirements for releasing the cost object cannot be met. In this case, d.velop smart invoice displays a corresponding error message.

The release sequence is thus as follows: First, the person identified in step 2 releases, and this is followed by the release sequence from step 1. If no person could be identified in step 2, the release sequence from step 1 applies. The person identified in step 3 then releases.

Example 4

There is an invoice for EUR 12,000.00 for cost center 2000. For amounts of EUR 10,000.00 and up, two persons must release the document.

Person	Release limit in EUR	Cost center
A	5.000,00	2000
B	20.000,00	2*
C	4.000,00	2000
D	50.000,00	

This invoice is sent to person A and then to person B for release.

- The **bottom to approver** release strategy determines a release sequence that already includes a sufficient number of people to meet the release requirement.
- Person A shares the highest specificity with person C, but has the highest release limit below the invoice amount.
- Person B has the next lowest specificity and also a sufficient release limit. The release process ends once the release has been granted.

Example 5

There is an invoice for EUR 12,000.00 for cost center 2000. For amounts of EUR 10,000.00 and up, two persons must release the document.

Person	Release limit in EUR	Cost center
A	13.000,00	2000
B	20.000,00	2*
C	4.000,00	2000
D	50.000,00	

This invoice is sent to person C and then to person A for release.

- The **bottom to approver** release strategy determines a release sequence that does not include a sufficient number of people to meet the release requirement.
- Person A has the highest specificity and also a sufficient release limit.
- Since a second release is required, a person is selected from the matrix who matches the document in terms of their specificity and whose release limit is the highest below the amount to be released. Person C meets these conditions. Person C is inserted at the start of the existing release sequence.

Skipping the release

You can skip a release step with release matrix if reviewers are authorized to release the document according to the release matrix.

Suppose you have a workflow definition with a **Verification** step and a subsequent **Release** step with release matrix. The action of the type **Confirm** in the **Release** step is **Release invoice** and leads to the **Accounting** step. Some people who perform the factual verification are represented by entries in the release matrix and are therefore authorized to release certain cost objects. If a person has performed the factual verification and is also authorized to release, the release step is to be skipped and the workflow is to continue directly with the **Accounting** step.

This is how it works

1. In the d.velop smart invoice administration interface, navigate to the **Workflow sequence**.
2. Open the relevant workflow definition.
3. Navigate to **Steps > Release**.
4. Switch to **Actions**.

5. Under **Release invoice**, click **Edit**.
6. Activate **Default action for this step** and click **Update**.
7. Switch to **General**.
8. Activate **Automatically create approvals from verifications**.
9. Save the workflow definition.

If a person performs the factual verification and is also authorized to release, a release will be automatically created for that person from the verification. If no further release is required for the document, the release step will be skipped.

Example

Person	Release limit in EUR	Cost center
A	5.000,00	2000
B	5.000,00	2000
C	5.000,00	2000
D	6.000,00	2000

There is an invoice in the amount of EUR 4,000.00 for the cost object with cost center 2000. Person A performs the factual verification and completes the **Verification** step. The workflow skips the release step because person A is also authorized to release and no further release is required. An automatic release is created for person A from the verification and logged. The workflow continues with the default action of the **Release** step. Therefore, the next step is **Accounting**.

Note

If the dual control principle is activated, a release step cannot be skipped. Furthermore, no automatic release is created for a person from the verification in this case.

Examples of combining features

Sending a document to multiple recipients and "bottom to approver" release strategy

There is an invoice for EUR 25,000.00 for cost center 2000.

Person	Release limit in EUR	Cost center
A	20.000,00	2000
B	8.000,00	2000
C	30.000,00	20*
D	20.000,00	200*
E	20.000,00	200*

This invoice would first be sent to person A, then persons D and E, and then person C for release.

Sending a document to multiple recipients, skipping releases, and "bottom to approver" release strategy

Example 1

There is an invoice for EUR 25,000.00 for cost center 2000. Person A has already performed the verification and the dual control principle is not activated.

Person	Release limit in EUR	Cost center
A	15.000,00	2000
B	8.000,00	2000
C	30.000,00	20*

Person	Release limit in EUR	Cost center
D	20.000,00	200*
E	20.000,00	200*

This invoice would first be sent to persons D and E, and then to person C for release. An automatic release is created for person A from the verification.

Example 2

There is an invoice for EUR 25,000.00 for cost center 2000. Person C has already performed the verification and the dual control principle is not activated.

Person	Release limit in EUR	Cost center
A	15.000,00	2000
B	8.000,00	2000
C	30.000,00	20*
D	20.000,00	200*
E	20.000,00	200*

This invoice would first be sent to person A and then to persons D and E for release. An automatic release is created for person C from the verification. With the **bottom to approver** release strategy, all specificity levels from most specific to less specific must be passed through and involved in the release process until the document is released by the first authorized person with the relevant release limit. Although the document has already been released by the person authorized to release based on the amount limit (person C), the more specific persons will still receive the document for release according to the **bottom to approver** release strategy.

Sending a document to multiple recipients and forced dual control principle

There is an invoice for EUR 15,000.00 for cost center 2000. For amounts of EUR 10,000.00 and up, two persons must release the document.

Person	Release limit in EUR	Cost center
A	20.000,00	2000
B	8.000,00	2000
C	30.000,00	2000
D	20.000,00	2000
E	20.000,00	200*

This invoice would first be sent to person B and then to persons A and D for release. Even if the document is sent to multiple recipients, it only needs to be released by one person. Therefore, the forced dual control principle requires person B to be involved in the release.

1.3.5. Configuring fields

Each document in d.velop smart invoice has a fixed set of fields in the document header and at the line item level. You can select which of these fields you want to use for a particular workflow. Since it is often necessary during the workflow to enter invoice data that are not covered by the fields provided, there are also custom fields provided in the document header and the line items. You can rename these fields and use them to enter additional posting data such as project numbers, vehicle license plates, etc.

Setting up a custom field

A total of twenty custom fields are available in the document header and in the line items. These fields are named **Custom 1** to **Custom 10** or **Custom 20** by default. You can rename the fields as you wish for each workflow.

Suppose you want to rename the **Custom 1** field in the line items to **Project number**.

This is how it works

1. Navigate to the field settings.
2. Switch to the **Custom fields** perspective.
3. Navigate to the **Line items** section and check the box for **Custom 1**.
4. Enter the **Project number** in the field labeled **en**.
5. Enter any translations in the corresponding language fields.
6. Save the workflow configuration.

If included in the particular layout, the field will be displayed in the workflow.

Renaming a field

In some situations it may be useful to rename a field. This may be the case, for example, if there is a field with the same function but a different name in the connected ERP system.

Suppose that the ledger account is named **Title** in the ERP system. To make it clearer for users, you want to rename the **Ledger account** field in the workflow to **Title** as well.

This is how it works

1. Navigate to the field settings.
2. Switch to the **Rename fields** perspective.
3. Find the **Line items** section and check the box for **Ledger account**.
4. Enter the **Title** in the field labeled **en**.
5. Enter any translations in the corresponding language fields.
6. Save the workflow configuration.

The **Ledger account** field will now be displayed in the workflow as **Title**.

1.3.6. Configuring the layout

You can freely configure the available fields in the workflow and their layout. There is a global layout that is used for all steps for which have no specific layout.

Adjusting the header layout

The fields in the header area are arranged in a grid. Each row of this grid has twelve columns for the fields and the corresponding labels. The header area can have any number of rows.

You can configure where the fields are located in these rows and columns.

This is how it works

1. Navigate to the workflow settings.
2. Switch to the **Layout** perspective.
3. In the header area you can drag and drop new fields from the list at the top into free boxes in the grid. You can reorder existing fields by dragging and dropping them within the grid.
4. Use the gear icon to adjust the width of the field or to remove the field from the table.

Adjusting the layout of the line item table

You can adjust the order and widths of the fields in the line item table to suit your preferences.

This is how it works

1. Navigate to the workflow settings.
2. Switch to the **Layout** perspective.
3. You can add new fields by dragging and dropping them into free areas at the beginning or end of the list. You can reorder existing fields by dragging and dropping them.

4. Click the gear icon to adjust the width of the field or to remove the field from the table.

Specifying whether fields are visible and editable

For each step you can specify whether a field is visible, editable, or mandatory. At the line item level, the option **Writable if verified** is still available, which makes a field editable even if the line item has already been verified. This can be useful if you want the account assignment to still be editable during a final check by the accounting department.

Suppose you want the fields **Vendor** in the header and **Order no.** in the line items to be both visible and editable.

This is how it works

1. Navigate to the settings of the corresponding step.
2. Switch to the **Fields** perspective.
3. If the **Vendor** field does not already exist in the **Header** area, add the field to the list by clicking **Add more fields**.
4. In the **Header** area, in the row for the **Vendor** field, select the **Visible** and **Writable** check boxes.
5. If the **Order no.** field does not already exist in the line items area, add the field to the list by clicking **Add more fields**.
6. In the **Line items** area, in the row for the **Order no.** field, select the **Visible** and **Writable** check boxes.

Note

Some configurations for the field properties appear useful at first glance, but are actually problematic.

Example

The **Verified** field fulfills multiple functions: When users activate the corresponding check box, they confirm the factual correctness of a line item. The field also represents the PO-related correctness of a line item with PO reference. If a line item matches the PO, the configuration of **Verified** as a mandatory requirement prevents the workflow from continuing.

If you want to ensure that the factual verification is performed for all line items, use the **voucher.verified?** condition. If you want to ensure that the factual verification is performed for all line items or that the line items are considered correct with regard to the PO, use the **voucher.matches_po_or_verified?** condition. For more information, see [Information about objects and fields](#).

1.3.7. Setting up formal and factual verification

During formal verification, users are tasked with verifying the formal correctness of the invoice: Is all the necessary information provided? Have the tax rates for the rendered services been correctly selected and reported by the supplier?

The formal verification is executed by a property of a connection between two steps. When you select the connection in question, the formal verification is issued. This makes it possible to set up a separate step for the formal verification with a separate assignee. However, the formal verification can also be executed as an additional task within another step, e.g. the factual verification.

To set up a verification, you have to make two configurations in d.velop smart invoice:

- Specify which users and groups can be selected as factual reviewers.
- Ensure that the **Verification** step is assigned to the selected reviewers.

Assigning the "Formal verification" attribute using a connection

To mark a document as arithmetically (formally) correct, you must configure a connection that, when selected, sets the **Formal verification** flag for the connection. The time and the user who carried out the operation are logged.

Suppose you want the formal approval to be done in the **Verification** step. If the invoice is error-free, the verification is followed by the **Release** step. If the invoice is incorrect, the **Cancellation** step follows.

This is how it works

1. Navigate to the settings for the **Verification** step.
2. Switch to the **Actions** perspective.
3. Navigate to the connection to the **Release** step and open the settings by selecting the **Edit** item from the connection's action menu.
4. Under **Formal verification**, select the option **Grant**.
5. Click **Save**.

Setting up a user finder for the "factual verification" step

In order for the **Verification** step to be assigned to the users and groups selected as factual reviewers, you must configure an appropriate user finder in this step.

This is how it works

1. Navigate to the **Verification** step and create a new user finder.
2. Give the user finder the title **Verification assignee**.
3. Enter the following Urkel expression under **Condition**: `user.name IN voucher.outstanding_verifiers`.
4. Select the **Use all hits** check box.
5. Click **Save**.

Selecting the set of factual reviewers

Suppose you want all groups starting with **FR_** to be selectable.

This is how it works

1. Navigate to the workflow settings.
2. Switch to the **Verification** perspective.
3. Create a new user finder using the button with the plug symbol.
4. Give the user finder a suitable name, for example **Verification**.
5. Enter the following Urkel expression under **Condition**: `/^SP_.*$/ ~ = user.name`
6. Click **Save**.

1.3.8. Setting up advanced verification

The classical verification in d.velop smart invoice is a simple check at the line item level. At the line item level, exactly one factual reviewer can be manually defined to handle the verification for the workflow.

Note

The advanced verification does not work in combination with the automatic purchase order comparison.

With the advanced verification, you can define any number of verification types, which will be run through one after another. In the workflow, not all defined verification types have to be performed for each line item. Verification types can be skipped, but at least one verification type is mandatory. You can manually select reviewers for each verification type and line item. To do this, you define user finders for each verification type. In addition, a mechanism similar to the release matrix can be used to preassign reviewers for the various verification types.

Creating the verification types

Suppose you want to create the appropriate verification types for the advanced verification.

This is how it works

1. Navigate to your workflow and click **Settings**.
2. Click the **Verification** tab.
3. Add at least one new verification type using the **Add verification** field.
4. For **Verification type**, enter the name of your verification.
5. For **Verifier**, save the user finder for this verification type.
6. Switch the verification type to **Active**.
7. Repeat the previous steps for each additional verification type.
8. You can use drag and drop to change the order of the individual verification types.

Adding a verification step to the workflow

Next, you need to add a verification step to the workflow. Only one verification step can be created in a workflow at a time.

This is how it works

1. In the d.velop smart invoice administration interface, navigate to your workflow and click **Steps** and then **New step**.
2. Give the step a title and select **Verification** as the step type. You can also use a verification matrix to automatically assign reviewers to verification types.
3. In your workflow, add the **Verified?** and **Verification** fields. These replace the old fields **Verifier** and **Verified**.

1.3.9. Useful information about the clearing house

The clearing house allows authorized users to intervene in the invoice workflow in the event of an error. Examples of errors include:

- A misdirected invoice
- Incorrect account assignment
- A mistakenly granted release
- An error in the verification

Authorized users can put an affected invoice into clearing mode and redirect an invoice or initiate a new verification or release. All other users have the option to actively send an invoice into clearing if they believe there is an error.

In clearing mode all fields are writable. All changes made within the clearing house are documented in the workflow log.

Authorized users can activate clearing mode only for invoices whose workflow is in a workflow step with user interaction. If the workflow of an invoice is in a workflow step or in a workflow phase without user interaction, users cannot activate clearing mode for this invoice.

Workflow steps without user interaction:

- **Wait for goods receipt** (Note that authorized users can cancel the **Wait for goods receipt** workflow step. In this case, the workflow is continued in a workflow step with user interaction.)

Workflow phases without user interaction:

- **Workflow started**
- **Workflow start failed**

- Workflow has an active document data transfer
- Workflow is being terminated/canceled
- Workflow has been terminated/canceled

Setting up the clearing house

To be able to use the new clearing house in the workflow, it may be necessary to activate a feature flag.

This is how it works

1. Navigate to your workflow definition in the d.velop smart invoice administration interface and click **Settings**.
2. Click the **Clearing** perspective.
3. Activate the **Clearing** option.

Editing documents in clearing mode

In this step, you must authorize at least one group to edit documents in clearing mode.

This is how it works

1. Navigate to your workflow definition in the d.velop smart invoice administration interface and click **Settings**.
2. Click the **Clearing** perspective.
3. With permissions set to **Default for all**, select a group to receive the tasks under **Group**.

Authorizing individual groups for each company

For each company, you can authorize individual groups to process documents in clearing mode.

This is how it works

1. Navigate to your workflow definition in the d.velop smart invoice administration interface and click **Settings**.
2. Click the **Clearing** perspective.
3. Under **Permissions**, click **New permission**.
4. Enter a company ID.
5. Select the appropriate group for this company under **Group**.

1.3.10. Configuring buckets

You can create any number of master data sets or "buckets." You can manage this master data via the administration interface or deliver it via the API. There are different usage scenarios for buckets:

- Regular export of master data from an ERP system via the bucket API.
- Manual maintenance of individual master data for entities that do not exist in the ERP system (for example, departments).

Creating a new bucket

Suppose you want to create a new master data bucket.

This is how it works

1. Navigate to the **Buckets** section.
2. Click **New bucket**.
3. Choose a name for the bucket, e.g. **ERP master data**.
4. Click **Save**.

Editing the data in a bucket

You can view and edit the data in a bucket at any time. Suppose you want to edit the data in the **ERP master data** bucket.

This is how it works

1. Navigate to the **Buckets** section.
2. Click the action menu next to **ERP master data** and choose **Edit data**.
3. Click **New company**. You can edit and delete existing datasets using the action menu for each tenant.
4. Click **Clients** next to **Choose Entity** to access other entities such as accounts payable or ledger accounts.

Importing sample data

Suppose you want to fill the **Demo** bucket with sample data so that you can get started quickly.

This is how it works

1. Navigate to the **Buckets** section.
2. Click the action menu next to **Demo** and choose **Import demo data**.

The bucket is now filled with sample data.

Using a bucket in a workflow

To use the data from a bucket in a workflow, you have to create an integration and link the fields of the workflow to that integration. Suppose you want to use accounts payable and ledger accounts from the **ERP master data** bucket.

This is how it works

1. Navigate to the workflow settings.
2. Switch to the **Integrations** section.
3. Navigate to **New integration > Bucket**.
4. Enter a name for the integration, for example **Master data**.
5. In the **Bucket** picklist, select the **ERP master data** bucket.
6. Enable the integration.
7. Click **Save**.
8. Switch to the **Fields** section.
9. In the **Vendor** row of the **Header** area, select the **Master data** data source.
10. In the **Ledger account** row of the **Line items** area, select the **Master data** data source.
11. Save the configuration.

The **Vendor** and **Ledger account** fields can now use the data from the **ERP master data** bucket.

1.3.11. Configuring integrations

You can use integrations to connect ERP systems. An integration can both transfer document data to an ERP system and provide master data, for example, accounts payable, ledger accounts or purchase orders.

d.velop smart invoice contains several standard integrations that you can use to connect general ERP systems. In addition, there are ready-made integrations for some ERP systems.

Creating a new integration

Suppose you want to create a new bucket-type integration.

This is how it works

1. Navigate to the workflow settings.
2. Switch to the **Integrations** section.
3. Navigate to **New integration > Bucket**.
4. Enter a name for the integration, for example **Master data**.
5. Configure the integration.
6. Make sure that the integration is active.
7. Save the configuration.

You can now use the integration.

Connecting a field to an integration

If you want to use data from an integration as a value list for a field, then you must connect the field to the integration. Suppose you want to connect the **Vendor** field to the **Master data** integration.

This is how it works

1. Navigate to the workflow settings.
2. Switch to the **Fields** section.
3. In the **Vendor** field of the **Header** area, select the **Master data** integration.
4. Save the configuration.

The **Vendor** field now allows you to select vendor data from the **Master data** integration.

Transferring document data to an integration

You can transfer document data to an integration whenever the invoice moves from one workflow step to another.

Suppose you want the document data to be passed to the **ERP webhook** integration when the workflow is completed from the last **Accounting** step.

This is how it works

1. Navigate to the workflow settings.
2. Switch to the settings for the **Accounting** step.
3. Switch to the **Actions** perspective.
4. Open the settings for the connection that ends the workflow.
5. Select the **ERP webhook** integration under **Run Export**.
6. Save the configuration.

Specifying an export value during the transfer to the ERP system

You can add an export value to the JSON object in the last step of the transfer to the ERP system. The addition does not have an effect on d.velop smart invoice.

This is how it works

1. Navigate to the corresponding workflow in the d.velop smart invoice administration area under **Workflow sequence**.
2. Navigate to **Steps** and select the relevant step.
3. Navigate to **Actions** and click on the menu with the three horizontal dots in the corresponding action and select **Edit**.
4. Select the export value to be transferred to the ERP system with the JSON object. You also have the following options: **default**, **validate** and **book**.

The selected export value is then integrated into the JSON object.

1.3.12. Working with due dates

You can display detailed information about the due date for an invoice. To visualize payment due dates, you can configure a traffic light system.

d.velop smart invoice displays the due date status for your documents in traffic light colors. If the number of days remaining until a document is due falls below the values defined here, the traffic light color changes. For the colors yellow and red, you define the number of days remaining until the document is due. If there are more days left than you specified for the color yellow, the traffic light will show green. However, if the due dates have been exceeded, the traffic light shows dark red.

d.velop smart invoice automatically calculates the due dates, any cash discount amounts and the payment amounts based on the selected payment terms, the gross invoice amount and the invoice date.

Note

Notes on calculating the due dates

When calculating the due dates, the values for the discount period and net period of a payment term are used.

Both the discount period and the net period can have the value 0 (days) or no value (NULL) in the master data. The values are interpreted differently as follows:

If the discount period of a payment term is 0 days, the discount period is interpreted as due on the invoice date. The discount due date therefore corresponds to the invoice date and is taken into account in the display of due dates.

If the discount period of a payment term is not specified (NULL), the payment term is interpreted as without discount term. No discount due date is shown in the display of due dates.

If the net period of a payment term is 0 days, the net period is interpreted as due on the invoice date. The net due date therefore corresponds to the invoice date and is taken into account in the display of due dates.

If the net period of a payment term is not specified (NULL), the payment term is interpreted as without net term. No net due date is shown in the display of due dates.

In the workflow dialog and incoming invoice list, users are given detailed information about when the net or discount due date for the invoice is reached.

Activating due dates

This is how it works

1. Navigate to the **Workflow sequence** administration.
2. Navigate to **Settings > Due dates**.
3. Click **Due dates**. You are provided with a suggestion for the traffic light system configuration.

You can then configure the traffic light system for the net and discount due dates.

1.3.13. Working with escalations

d.velop smart invoice provides various escalation mechanisms to help users meet deadlines.

Escalation with fixed deadline

If a task remains in the same step, e.g. verification, for more than x calendar days, an escalation is triggered.

Escalation with dynamic deadline

Enter a deadline based on the payment terms in the document. Example: For example, if the payment terms specify a net period of 30 days and a discount period of 10 days, you can set an escalation to occur three days before the discount period expires and five days before the net period expires.

If an escalation is triggered, a person determined by a user finder will receive a notification.

Note

Weekdays or work days

You can configure whether the escalation deadlines should be evaluated in weekdays or work days. Work days include the weekend (Saturday, Sunday) as non-work days. (Regional) holidays or other non-work days cannot be taken into account.

There are three different types of escalations:

1. The global first-stage escalation.
2. The step-specific first-stage escalation. If this escalation is configured, the global first-level escalation is no longer taken into account.
3. The global second-stage escalation. This escalation is always taken into account and can only be configured globally.

Configuring a global escalation with dynamic deadline

An escalation with a dynamic deadline is always based on payment terms. In order to use the deadline, the document must contain a payment term. Based on the payment terms and the invoice date, the software calculates the time points at which the net and cash discount periods expire for the document. You can specify how many days before these time points an escalation will be triggered.

Note

Note on calculating escalation times

When calculating the escalation times, the values for the discount period and net period of the selected payment term are used. Both the discount period and the net period can have the value 0 (days) or no value (NULL) in the master data. Both values are interpreted identically as follows:

- Documents with a payment term where the discount period is 0 days or not specified (NULL) do not escalate if dynamic escalation periods are defined before the expiry of the discount period, otherwise they would escalate immediately.
- Documents with a payment term where the net deadline is 0 days or not specified (NULL) do not escalate if dynamic escalation periods are defined before the net deadline expires, otherwise they would escalate immediately.

Suppose you want to trigger an escalation four days before the net deadline is reached and two days before the cash discount deadline is reached, regardless of which step the workflow is currently in. You want the escalation notifications to be delivered to the **Accounting** group.

This is how it works

1. Navigate to the **Workflow sequence** administration.
2. Click the desired workflow definition.
3. Navigate to the workflow settings.
4. Switch to **Escalation**.
5. Create a new deadline using the button next to the **Deadline** field in the **Escalation 1** section.

6. Enter a name for the deadline in the **Title** field.
7. Enter the value **2** in the **Days before cash deadline** field.
8. Enter the value **4** in the **Days before net deadline** field.
9. Optional: Activate the option **Count escalation days in work days (Mon-Fri)**.
10. Click **Add**.
11. Click the button next to the **User finder** field in the **Escalation 1** section.
12. Enter the value **Accounting** in the **User finder** field.
13. Enter the value **user.name = "Accounting"** in the **Condition** field.
14. Click **Add**.

Configuring an escalation with fixed deadline for a step

An escalation with a fixed deadline is based on the time the current step was delivered to the assignee. You can specify how many days after delivery the escalation will be triggered.

Suppose you want to configure an escalation after three days for the **Release** step. You want the escalation notifications to be delivered to the **Accounting** group.

This is how it works

1. Navigate to the **Workflow sequence** administration.
2. Click the corresponding workflow definition.
3. Click on the **Steps** perspective and select the **Release** step.
4. Switch to **Escalation**.
5. Create a new deadline using the button next to the **Deadline** field.
6. Enter a name for the deadline in the **Title** field.
7. Enter the value **3** in the **Fixed days after receiving task** field.
8. Optional: Activate the option **Count escalation days in work days (Mon-Fri)**.
9. Click **Add**.
10. Click the button next to the **User finder** field.
11. Enter the value **Accounting** in the **User finder** field.
12. Enter the value **user.name = "Accounting"** in the **Condition** field.
13. Click **Add**.

1.3.14. Working with goods receipts

With goods receipts you record the receipt of ordered goods in your ERP system. You can provide goods receipts in d.velop smart invoice for use in the verification and release workflow. The following options are available for assigning goods receipts to an invoice in a workflow:

- If document lines with referenced PO line items and delivery note numbers for document lines exist when a workflow is started, a corresponding goods receipt line is automatically assigned.
- The workflow step **Wait for goods receipt** automatically assigns suitable goods receipt lines to existing document lines with a referenced PO line item. The workflow step **Wait for goods receipt** takes place without user interaction.
- You can then manually assign corresponding goods receipt lines to an existing document line with a referenced PO line item.

Using goods receipts

Configure the purchase order comparison to be able to use goods receipts in d.velop smart invoice.

Suppose you want to use goods receipts in addition to POs in d.velop smart invoice.

This is how it works

1. Navigate to the **Workflow sequence** administration.
2. Click the corresponding workflow definition.

3. In the workflow settings, navigate to **General**.
4. Navigate to **Purchase order matching > Purchase orders/goods receipts**.
5. Select **POs and goods receipts**.

The workflow step “Wait for goods receipt”

Digital invoice delivery often means that an invoice is received before the goods invoiced even arrive at the warehouse.

An invoice for which not all goods receipts are available would have to be manually resubmitted on a regular basis in order to check the goods receipts. To avoid this manual resubmission, you can configure the workflow step **Wait for goods receipt**.

The workflow step **Wait for goods receipt** regularly checks workflows with the status **Waiting for goods receipt**, unless there are reasons to the contrary, and automatically assigns suitable goods receipt lines to the document lines with a referenced PO line item.

Configuring the workflow step “Wait for goods receipt”

Configure the workflow step **Wait for goods receipt** to automatically assign matching goods receipt lines to existing document lines with a referenced PO line item.

Suppose you want the workflow step **Wait for goods receipt** to wait a maximum of three days for goods receipts. You also want to allow the user group **Purchasing** to manually complete the workflow step.

This is how it works

1. Navigate to the **Workflow sequence** administration.
2. Click the corresponding workflow definition.
3. Navigate to **Steps** and click **New step**.
4. Enter a title for the step under **Title**.
5. Optional: In the language-dependent fields under **Translation**, enter a title for the step in the respective language.
6. Under **Step type**, select the type **Wait for goods receipt**.
7. In the **Assignee** section, click the button next to **User finder**.
8. Under **User finder**, enter the value **Purchasing**.
9. Enter the value **user.name = "Purchasing"** under **Condition**.
10. Click **Add**.
11. In the **Wait for goods receipt** section, click the button next to **Deadline**.
12. Enter a title for the waiting period under **Title**.
13. Under **Fixed days after receiving task**, enter the value **3**.
14. Optional: Activate the option **Count escalation days in work days (Mon-Fri)**.
15. Click **Add**.

You have configured the workflow step **Wait for goods receipt**.

You must perform further configuration to use the new workflow step in your workflow sequence.

Note

Note on the waiting period

The waiting period configured under **Deadline** only takes into account a specified deadline. This depends on whether a payment term is included in the document and, if so, which payment term. Suppose the document waiting for goods receipts has a payment term with a discount and a net payment period. In addition, the waiting period for the workflow step **Wait for goods receipt** has positive values for both **Days before cash deadline**, **Days before net deadline** and **Fixed days after receiving task**. In this case, the waiting period for the workflow step **Wait for goods receipt** is determined based on the value **Days before cash deadline**.

In general:

- If a waiting period has a value for **Days before cash deadline** and the payment term configured in the document has a discount period, the waiting period is calculated after the discount period.
- If no value is entered for **Days before cash deadline** or the payment term configured in the document does not have a discount period: If the waiting period has a value for **Days before net deadline** and the payment term specified in the document has a net deadline, the waiting period is calculated after the net deadline.
- If no value is entered for **Days before net deadline** or the payment term configured in the document does not have a net deadline: The waiting period is calculated according to the value **Fixed days after receiving task**.

Using the “Wait for goods receipt” workflow step

Suppose you want the workflow to continue with the workflow step **Invoice verification** after successfully assigning matching goods receipt lines to all document lines with a referenced PO line item. If no goods receipt lines have been assigned, the workflow should continue with the workflow step **Invoice verification (warehouse/goods receipt)**.

This is how it works

1. In the previously created workflow step **Wait for goods receipt**, navigate to **Actions**.
2. Click **New action**.
3. Enter a title for the action under **Title**.
4. Under **Type**, select **Confirm**.
5. Click **Add**.
6. Click **New connection**.
7. Enter a name for the connection under **Name**.
8. Under **To**, select the target step **Invoice verification**.
9. Enter the value **true** under **Condition**.
10. Click **Add**.
11. Repeat steps 1 to 10. In step 4, select **Reject** as **Type**. In step 8, select **Invoice verification (warehouse/goods receipt)** as the target step.

Warning

The workflow step **Wait for goods receipt** is a step without user interaction, meaning that there is also no user context. Therefore, do not configure any outgoing step connections that require a user context. Otherwise, it will not be possible to exit the **Wait for goods receipt** step. The following configurations are not permitted:

- **Grant approval or revoke approval**
- **Formal verification, Grant or Formal verification, Revoke**
- **Remove verification**
- **Comment required**

How the workflow step "Wait for goods receipt" works

The workflow step **Wait for goods receipt** works according to a clear flow chart.

When a workflow enters the step **Wait for goods receipt**, the workflow is transferred to background processing. In background processing, a check is carried out at a fixed interval from the time the step is started to see whether there are matching goods receipt lines for the document lines with a referenced PO line item. The check runs for the configured waiting period as a maximum.

Note

There are scenarios in which the first iteration of the workflow step **Wait for goods receipt** can already assign a matching goods receipt line to all document lines with a referenced PO line item, for example if there is exactly one document line with a referenced PO list item for which there is exactly one matching goods receipt line. The processing time of the step **Wait for goods receipt** can vary for different workflows with this scenario (from a few moments to an hour). The reason for different processing times is usually due to higher loads in the background processing. If the job queues in background processing are long, the processing of individual jobs takes correspondingly longer. In principle, however, processing should not take longer than a few minutes.

Checking whether the workflow meets the requirements to wait for goods receipts

Before the step **Wait for goods receipts** determines and assigns suitable goods receipt lines to document lines with referenced PO line items, the step **Wait for goods receipts** checks whether the workflow meets all the requirements to be able to wait for goods receipts. The following aspects are examined:

Is the workflow still in the "Wait for goods receipt" step?

As soon as a job for a workflow that is waiting for goods receipt is processed in the background, it is first checked whether the workflow is still in the **Wait for goods receipt** step.

Example: For a workflow, a job has been placed in the **Wait for goods receipt** queue. Before the job is processed, an authorized user manually cancels the workflow step **Wait for goods receipt**. The job remains in the queue. When the job is processed, a check is carried out to see whether the workflow associated with this job is still in the **Wait for goods receipt** step. If the workflow is not in the **Wait for goods receipt** step, the job will terminate without result.

Is the document in the workflow not a duplicate or an accepted duplicate?

If a workflow is still waiting for goods receipts, the duplicate check is carried out if you have activated the duplicate check in the workflow definition. The duplicate check checks whether the document in the workflow could be a duplicate. If the duplicate check detects a possible duplicate, it checks whether the document was accepted as a duplicate. If the document was not accepted as a duplicate, the workflow step **Wait for goods receipt** is exited using the configured **Reject** action. Carrying out a duplicate check as part of the **Wait for goods receipt** routine is particularly useful if **Wait for goods receipt** is the first workflow step in a workflow definition.

Is the document linked to a PO in the workflow?

If a workflow is still waiting for goods receipts and the duplicate check was successful, a check is carried out to see whether the document in the workflow references a PO. If the document is not linked to a PO, it is not possible to wait for goods receipts. The workflow step **Wait for goods receipt** is exited using the configured **Confirm** action. A **Problem** event is then created.

Does the document have at least one document line that is linked to a PO line item?

As well as being linked to at least one PO, the document in the workflow must have at least one document line with a referenced PO line item in order to be able to wait for goods receipts. If there is not at least one document line with a referenced PO line item, the workflow step **Wait for goods receipt** is exited using the configured **Reject** action. A **Problem** event is then created.

Is the workflow still within the defined waiting period?

You must define a waiting period for the workflow step **Wait for goods receipt**. A workflow waits for goods receipts for the specified waiting period as a maximum. If the defined waiting period is exceeded, the workflow step **Wait for goods receipt** is exited using the configured **Reject** action. A **Problem** event is then created.

If all the prerequisites described above are met for a workflow, the workflow step **Wait for goods receipt** starts by assigning suitable goods receipt lines to document lines.

Determination and allocation of suitable goods receipt lines

The workflow step **Wait for goods receipt** goes through the steps described below. The steps take place for each document line with a referenced PO line item.

Step 1: Check whether the goods receipt line must be assigned to the referenced PO line item on the document line

In the master data of a PO line item, you can specify whether the assignment of a goods receipt line is mandatory. If the assignment is mandatory for the PO line item referenced in the document line, the workflow step tries to determine a suitable goods receipt line.

Note

What if the assignment of a goods receipt line is not mandatory?

If the assignment of a goods receipt line is not mandatory, the workflow step checks the PO line item referenced in the document line to determine whether the quantity received is sufficient for the quantity invoiced with this document line. The workflow step also takes into account the document lines of other running or completed workflows that reference the same PO line item. If the quantity received on the PO line item is not sufficient for the invoiced quantities of all referenced document lines, the system continues to wait.

Step 2: Check whether the document line has not yet been assigned to a goods receipt line

A search is only carried out for a suitable goods receipt line if a document line with a referenced PO line item has not yet been assigned a goods receipt line. If a goods receipt line has already been assigned, the document line is skipped in further iterations of the workflow step **Wait for goods receipt**.

Step 3: Determination of suitable goods receipt lines

The workflow step determines suitable goods receipt lines for a document line with a referenced PO line item, based on the following flow chart:

1. Select the goods receipt lines that refer to the PO line item referenced in the document line (linked via the ID of the PO line item).
2. Only select goods receipt lines if the company corresponds to the company specified in the document.

3. Only select goods receipt lines if the vendor corresponds to the vendor specified in the document.
4. Only select goods receipt lines if the quantity corresponds to the quantity in the document line.
5. Only select goods receipt lines if they have not yet been assigned to any other document line of a workflow that is still running or has been completed.
6. If the document line has a value in the field **Delivery slip number**: Only select goods receipt lines if the delivery slip number corresponds to the delivery slip number in the document line.

Step 4: Assignment of suitable goods receipt lines

Option 1: There is only one matching goods receipt line

If the workflow step was able to determine only one matching goods receipt line, the workflow step assigns this goods receipt line to the document line.

Option 2: There is no matching goods receipt line

If the workflow step could not determine a suitable goods receipt line, the workflow step continues to wait for goods receipts. The next iteration of the workflow step **Wait for goods receipt** searches again for matching goods receipt lines.

Option 3: There is more than one matching goods receipt line

If several goods receipt lines are eligible for the document line, the workflow step cannot decide which goods receipt line should be assigned. The workflow step **Wait for goods receipt** is exited using the configured **Reject** action. A **Problem** event is then created.

If the workflow step was able to successfully assign a goods receipt line to each document line with a referenced PO line item or, if the assignment of a goods receipt line is not mandatory, a sufficiently high quantity was received for the order item, the workflow step **Wait for goods receipt** is exited using the configured **Confirm** action.

Problems processing the workflow step "Wait for goods receipt"

If problems occur during processing of the workflow step **Wait for goods receipt**, users can view the problems in another workflow step with user interaction in a dedicated tab.

The following problems can occur when processing the workflow step **Wait for goods receipt**.

Problem	Cause
No PO number found on the document	The document should wait for goods receipt. Processing did not find a PO number on the document. Therefore, the wait was ended.
No document line with reference to a PO found	The document should wait for goods receipt. Although there is a PO number on the document, processing has not found a document line that references a PO. Therefore, the wait was ended.
Waiting period expired	The document was waiting for goods receipt. During the waiting period, processing did not record sufficient goods receipts. Therefore, the wait was ended.
Various possible delivery slip assignments found	Processing found several possible delivery slip assignments for the lines on the document. Therefore, the wait was ended.

1.3.15. Setting up the transfer of line items from d.velop document reader invoice (on-premises)

Suppose you want to transfer line items that have been detected in d.velop document reader invoice to d.velop smart invoice. To do this, you must set up a multi-value property for the transfer of line items in the d.velop documents category that is used by d.velop smart invoice to save the invoices. You must then write the line item data from d.velop document reader invoice into this multi-value property, delimited by a separator. d.velop smart invoice recognizes the contents of the field when a new workflow is started and creates corresponding line items.

Connecting the new multi-value property with d.velop smart invoice

In order for d.velop smart invoice to read line item data from the new multi-value property, you must assign the multi-value property to the corresponding d.velop smart invoice field in the d.velop documents assignments.

Suppose you have created a multi-value property **Line item data**.

This is how it works

1. Navigate to the d.velop documents assignments.
2. Find the assignment for d.velop smart invoice **Invoices (source "smart invoice - invoice")**.
3. Add a new field.
4. Select **Import: line items** as the source field.
5. Select **Line items** as the target field.
6. Save the assignment.

Transferring line items from d.velop document reader invoice

Suppose you want to transfer line items from d.velop document reader invoice.

This is how it works

1. Configure d.velop document reader invoice such that the line items are written to the multi-value property you set up for this purpose in d.velop documents. d.velop smart invoice requires one line for each invoice line to be created.
2. Specify the fields for each line, separated by pipe characters (|). Enter the separators in direct succession (no space) if you do not want to transfer one of the fields. Then proceed in the following order:
 - Line number, mandatory, starting with 1
 - Net amount, with a point (.) as decimal separator
 - Gross amount, with a point (.) as decimal separator
 - Payment amount, with a dot (.) Decimal separator
 - Tax code, ID from ERP system
 - Purchase order (PO) number
 - PO line item
 - Quantity
 - Unit price, with a point (.) as decimal separator
 - Unit, for example piece
 - Price unit
 - Factual reviewer

Note

Example:

A line with a net amount of EUR 100.00, a gross amount of EUR 119.00 and the tax code EN_FULL:

```
1|100.00|119.00||EN_FULL
```

A line with a net amount of EUR 123.45 and a purchase order number and purchase order line item:

```
1|123.45||||PO0001|1
```

Specifying a factual reviewer

d.velop smart invoice requires the factual reviewer in the format **user:<id>** or **group:<id>**. Replace **<id>** with the ID of the user or group in d.ecs identity provider.

1.3.16. Performing a duplicate check

If the duplicate check is turned on, the software will check each invoice as it is processed to see if it has already been processed. If an invoice has already been processed, the software displays a warning message. Before you can continue processing the invoice, you must accept the warning and provide a justification. This justification is noted in the invoice log.

Enabling the duplicate check

For newly created workflows, the duplicate check is enabled by default. Suppose you want to enable the duplicate check for an existing workflow.

This is how it works

1. Navigate to the workflow settings.
2. Enable the **Duplicate check** in the section with the same name.
3. Save the workflow configuration.

The duplicate check is now enabled.

Detecting duplicates

The software marks a document as a duplicate under the following conditions:

- The document was processed within the last 18 months.
- The document was processed with an invoice from the same vendor (identified by the vendor number).
- The document was processed with the same external document number (invoice number).

1.3.17. Assigning workflows to repositories

For a workflow to function, you must assign the workflow to at least one repository where the invoice documents are stored.

Suppose you have configured a **Cost invoices** workflow that you want to assign to the repository **Productive archive**.

This is how it works

1. In the **Workflow sequence**, navigate to the overview of configured workflows.
2. Click **Connect repositories** from the context menu for the **Cost invoices** workflow.
3. In the table, select the **Assigned** option in the row for the **Productive archive** repository.
4. Click **Save**.

1.3.18. Information about conditions

In several places, you can define your own conditions that will react dynamically to values in the document. Examples include a user finder for a step or deciding which step to choose next. These conditions are specified in the Urkel expression language.

You can use Urkel expressions in various places. Depending on the use case, various objects are available according to the specific context. This chapter explains how the various entry points work and the objects available at each.

User finder

Urkel expressions in user finders are called once for each user and group. If an expression for a user or group returns the value **true**, the user or group is included in the result.

The software offers the following objects in user finders: **voucher, user, group, step**.

Conditions

Each connection between two steps has a condition specified as an Urkel expression.

The software offers the following objects in conditions: **voucher, step**.

Information about syntax, semantics and data types

Each Urkel expression has a data type. The available data types are:

- Boolean
- String
- Number (decimal number)
- List
- Regular expression

The expressions used must be of the Boolean type at the top level.

Simple Boolean expressions

The simplest Boolean expressions are the literals **true** and **false**. Boolean expressions can in turn be linked by the **AND** and **OR** operators. The following rules apply:

- true AND false results in false
- false OR true results in true

A preceding **NOT** can be used to negate a Boolean expression. **NOT true** results in false.

Comparing two expressions

Expressions of other types can be compared with each other. The operators **=, !=, <, >, <=, >=** are available for this purpose. Both operands must have the same type. The result of a comparison is Boolean. An example:

- "abc" = "def" results in false
- 5 > 7 results in false
- "def" != "def" results in false

Operations on lists

To be able to display multiple values, Urkel provides a built-in list type. Lists are sorted and can hold values of any type. Some properties of documents or users are of the **list** type. In Urkel expressions, lists can also be specified directly. The expression (true, false) results in a list with the values true and false. The expression (true, 1 < 2, 3) results in a list with the values true, true and 3.

If a list has only Boolean values as elements, the operators **ALL** and **ANY** can be used. **ALL l1** returns true if the list l1 contains only true values. **ANY l1** returns true if the list l1 contains at least one true value.

The length of a list l1 can be determined via **len(l1)**. The result is of the number type.

The **IN** and **NOT IN** operators can be used to check whether a value is contained or not contained in a list. An example:

- 1 IN (1, 2, 3) results in true
- 1 < 2 IN (3 > 4, 5 > 6, 7 > 8) results in false

Strings and regular expressions

The string type can contain strings of characters. Strings can be compared using the usual comparison operators. In addition, strings can be checked by a regular expression using the `=~` operator. The regular expression must be on the left side. An example:

- `/^ (Release|Accounting) \d+ $ / =~ "Release 1"` results in true
- `/\d{4} / =~ "123"` results in false

You can insert expressions into string literals and regular expressions using the `#{}` syntax. An example:

- `"Posting #{voucher.posting_text}"` results in "Posting Telecom" if the posting text is Telecom.
- `/^ Approver_# {voucher.company.nr} $ / =~ "Approver_23"` returns true if the company number is 23.

Accessing the properties of the document or user

All Urkel expressions are executed in a context where certain objects, such as the current user or document, are available. From Urkel expressions, you can access properties of these objects using point notation:

- `voucher.posting_text` returns the **posting text** property of the **Document** object.
- `voucher.company.nr` returns the **Nr** property of the **Company** property of the **Document** object.

The objects that are available in a particular context vary depending on the use case.

Information about objects and fields

This chapter explains all the objects available in Urkel expressions and their associated properties. Which objects are actually available varies depending on the use case.

Voucher (document)

Property	Type	Description
<code>doc_id</code>	String	The document ID in the repository
<code>company.nr</code>	String	Number of the company
<code>company.name</code>	String	Name of the company
<code>vendor.nr</code>	String	Number of the vendor
<code>vendor.name</code>	String	Name of the vendor
<code>vendor.email</code>	String	Vendor's e-mail address
<code>net_amount</code>	Number	Net amount of the invoice in cents
<code>net_amount_lcy</code>	Number	Net amount of the invoice in cents in local currency
<code>gross_amount</code>	Number	Gross amount of the invoice in cents
<code>gross_amount_lcy</code>	Number	Gross amount of the invoice in cents in local currency
<code>pay_amount</code>	Number	Payment amount of the invoice in cents
<code>pay_amount_lcy</code>	Number	Payment amount of the invoice in cents in local currency
<code>vat_amount</code>	Number	VAT of the invoice in cents
<code>vat_amount_lcy</code>	Number	VAT of the invoice in cents in local currency
<code>currency.id</code>	String	ID of the currency
<code>currency.code</code>	String	Code of the currency (e.g. EUR).
<code>currency.name</code>	String	Name of the currency
<code>barcode</code>	String	Barcode
<code>internal_number</code>	String	Internal document number
<code>external_number</code>	String	External invoice no.
<code>document_date</code>	String	Document date
<code>payment_date</code>	String	Payment date
<code>posting_period</code>	String	Posting period
<code>document_type.id</code>	String	ID of the document type
<code>document_type.name</code>	String	Name of the document type

Property	Type	Description
posting_text	String	Posting text
customN with N between 1 and 20	String	Custom fields
amount_allocated? (mode)	Boolean	<p>Is the header amount fully allocated to the line items?</p> <p>mode determines which data is checked:</p> <ul style="list-style-type: none"> • 0: Gross amount must match • 1: Net amount must match • 2: Payment amount must match • 3: Gross and net amounts must match • 4: Gross and payment amounts must match • 5: Net and payment amounts must match • 6: Gross, net and payment amounts must match
amount_allocated_tolerance? (mode, tolerance)	Boolean	<p>Is the header amount allocated to the line items within the tolerance?</p> <p>This function is useful when net or gross amount is to be divided exactly, but the other amount may differ minimally due to rounding differences.</p> <p>mode determines which data is checked. tolerance specifies the tolerance in cents.</p> <ul style="list-style-type: none"> • 0: Gross amount must match exactly, net amount may differ. • 1: Net amount must match exactly, gross amount may differ.
verified?	Boolean	<p>Are all document lines marked as factually verified?</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note</p> <p>A document line that matches the purchase order and is therefore considered factually correct will not be accepted by this condition. Document lines must have been marked as factually verified by a user.</p> </div>
matches_po?	Boolean	<p>Do all document lines match their respective PO lines?</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note</p> <p>A document line that has been marked by a user as factually verified will not be accepted by this condition. The document line must have been evaluated as correct based on the comparison with the purchase order data.</p> </div>
matches_po_or_verified?	Boolean	Do all document lines either match their respective purchase order lines or have been marked as factually verified?
verifiers	List of strings	A list of names of all users and groups that are entered as factual reviewers.
outstanding_verifiers	List of strings	A list of names of all users and groups that are entered as factual reviewers for line items that have not yet been marked as factually verified.
has_purchase_orders?	Boolean	Does the document have a PO reference?
has_goods_receipts?	Boolean	Are there any goods receipts on the document?
has_surcharges?	Boolean	Are there any surcharges on the document?
<property>.present? with <property> as placeholder for a property of the document object	Boolean	Is the property filled in?
<property>.blank? with <property> as placeholder for a property of the document object	Boolean	Is the property empty?

User

Property	Type	Description
name	String	Name of the user or group.
email	String	User's e-mail address. Not defined for groups.

Property	Type	Description
group_ids	List of strings	List of the IDs of all groups that the user belongs to. Empty for groups.
group_names	List of strings	List of the names of all groups that the user belongs to. Empty for groups.
id	String	The user's or group's ID from the source system (for example d.ecs identity provider).
type	String	Either user or group .

Step

Property	Type	Description
uid	String	The internal ID of the step.
title	String	The translated title of the step.
original_title	String	The title of the step if no translation is available.

1.3.19. Displaying the step comparison view

The step comparison view provides you with detailed information about document changes. This view displays any changes between the steps selected in the administration area.

This is how it works

1. In the d.velop smart invoice administration area, go to **Workflow sequence** and click the corresponding workflow.
2. Under **Steps**, select the workflow step in which you want to display the step comparison view.
3. Go to **Step comparison view**.
4. Activate the step comparison view.
5. Select a step for comparison.

The header and line item data is then annotated and compared with the workflow step you selected beforehand. The changes to the header and line item data are either displayed in red (deleted), yellow (changed) or green (added).

1.3.20. Useful information to know about transferring master data

In this chapter, you can learn all about transferring master data between d.velop smart invoice and d.velop document reader.

The master data is generally transferred from d.velop smart invoice to d.velop document reader. In the process, existing data in d.velop document reader is overwritten by the data from d.velop smart invoice.

Evaluating master data quality

d.velop document reader contains a function for evaluating master data quality. This evaluation function is also valid for the master data quality in d.velop smart invoice, because the master data is transferred from d.velop smart invoice to d.velop document reader. If the evaluation shows insufficient master data quality, check the synchronized bucket in d.velop smart invoice.

Synchronizing buckets

At present, only one bucket from d.velop smart invoice can be synchronized with d.velop document reader. If only one bucket has been created, this bucket is automatically selected for synchronization. If multiple buckets exist, you can go to the d.velop smart invoice administration and specify which bucket you want to synchronize.

This is how it works

1. Within the navigation bar, click **Recognition Backends** to go to the systems to which d.velop smart invoice transfers master data. An entry for d.velop document reader is created automatically.
2. Click the three horizontal dots in the row for d.velop document reader to enable editing.

3. Select the bucket to be synchronized. **Automatic** is set by default.

Transferring data

The following data is transferred from d.velop smart invoice to d.velop document reader:

- Clients with address information
- Vendors with address and tax information
- Vendor bank details
- PO data
- Goods receipts

Transferring master data from d.velop smart invoice to d.velop document reader

d.velop smart invoice transfers master data to d.velop document reader in the following scenarios:

- After each user interface change
- After an ERP import via API
- After data is retrieved from an external system (e.g. Microsoft Dynamics 365 Finance & Operations or Business Central Integration)
- After a manual transfer

The process of transferring master data is asynchronous and may therefore take several minutes.

1.3.21. Useful information to know about invoices with reference to a PO

Setting up invoices with reference to a PO lets you compare your received invoices with the applicable PO and transfer information from the PO to the line items. Below you can see a summary of the basic functions that d.velop smart invoice provides for invoices with a PO reference.

Useful information to know: general information and calculations

A PO consists of header and line item data. The PO number and information about suppliers, which you can find in the header data, are important pieces of information for processing invoices with PO references. The line items from the PO are also essential, because they list what was ordered.

Linking the PO line item and invoice line item

Each line has a PO number and a PO line item. Each invoice line item in d.velop smart invoice can be assigned to exactly one PO line item. For this purpose, each invoice line item has the fields **Order no.** and **Order line item**.

Useful information to know about the unit price, quantity, unit and price unit

Invoice line items contain fields such as **Item no.**, **Unit price**, **Amount** and **Unit**, for example. These fields can be used independently of POs, but primarily appear in lines with a PO reference.

In the simplest scenario, the **Unit price** and **Amount** fields determine the net price of the line with the PO reference through the following equation:

Unit price x quantity = net amount

The **Unit** field is also usually available for **Amount**. Common units include pieces, kilograms or meters, for instance. The unit in d.velop smart invoice is generally purely an information field that is not initially of relevance.

The behavior is different if the **Unit of price** (price unit) field is taken into account. A price unit is used whenever the unit price is specified for a unit of measure that differs from that of the amount. This is the case, for example, if the unit price in the PO and on the invoice is specified in "packaging per 100 pieces" but the amount is specified in individual pieces, because the item is to be managed in individual pieces in

the warehouse stock in the ERP system. To calculate the net price, the unit price must first be divided by the price unit before it is multiplied by the amount:

Unit price / price unit x quantity = net amount

Based on the example above, the price unit "100" would return the correct results, because the unit price is specified in 100 pieces but the amount is specified in individual pieces. The division calculates the unit price per piece.

PO master data from the ERP system

To work with POs in d.velop smart invoice, you require PO data. This generally comes from the existing active POs from the ERP system. This master data is provided through an integrating application. This often means that the data is regularly transferred to a bucket.

The master data contains the PO and the PO lines. A PO comprises information such as the PO number, a description and an (optional) status. A PO line contains information such as the item number, amount, unit price, price unit, tax code, cost center, and so on.

To work with POs, you must transfer these POs to the document from the master data. There are several ways to do so:

Retrieving PO lines

You can manually transfer the PO lines from the PO master. You can use the **Fetch order data** function to do so. You can use these functions to search through the PO master for POs for the selected companies and vendors. For each PO, you can display the available PO lines and choose which lines you want to transfer to the invoice. The amount ordered, delivered and calculated with each PO line item is also displayed.

Capturing PO lines with d.velop document reader

If you are using d.velop smart invoice in the cloud, d.velop document reader automatically detects line items on the invoice that are linked to POs. These line items are automatically transferred from the PO and provided with the values for prices and amounts that are found on the invoice.

Importing PO numbers

When you import a document to start a workflow, you can transfer a list of PO numbers. To do so, the relevant multi-value property must be assigned in d.velop documents. d.velop smart invoice determines the PO line items for which there is a remaining billable amount. This means that the PO master data has a delivered amount that is greater than the amount that was already processed by d.velop smart invoice. Please note that the values from the PO are transferred to the invoice in this scenario. No values for unit prices and amounts are read from the document.

Importing line items and POs

In addition to the option of importing a list of PO numbers, you can specify the line items to be created through a document property and establish the relationship to a PO line item. In this case, the data that is not transferred from the invoice line item is enriched with the PO master data. As a result, you can connect the results from an external document capture system (the amount and unit price, for example) to the PO functions in d.velop smart invoice.

1.3.22. Useful information about calculations in d.velop smart invoice

d.velop smart invoice performs calculations for different fields based on the data in the workflow. In the following sections, you will learn which fields can be calculated by d.velop smart invoice and how the calculations are performed.

Calculating the net amount of a line item

In the simplest scenario, the **Unit price** and **Amount** fields determine the net price of the line item through the following equation:

$$\text{Einzelpreis} \cdot \text{Menge} = \text{Nettobetrag}$$

In addition to the **Amount** field, the corresponding **Unit** field also exists in many cases. Common units include pieces, kilograms or meters, for instance. As a rule, the unit in d.velop smart invoice is purely an information field. The unit is not taken into account in the calculation of the net amount.

The behavior is different if the **Unit of price** (price unit) field is taken into account. A price unit is used when the unit price is specified for a unit of measure that differs from that of the amount. This is the case, for example, if the unit price in the PO and on the invoice is specified in “packaging per 100 pieces” but the amount is specified in individual pieces, because the item is to be managed in individual pieces in the warehouse stock in the ERP system. To calculate the net price, the unit price must first be divided by the price unit before the quotient is multiplied by the amount:

$$\frac{\text{Einzelpreis}}{\text{Preiseinheit}} \cdot \text{Menge} = \text{Nettobetrag}$$

Based on the example above, the price unit “100” would return the correct results, because the unit price is specified in 100 pieces but the amount is specified in individual pieces. The division calculates the unit price per piece.

Discounts

The net amount of a line item can also be influenced by any discounts granted.

Seven discount fields are available in d.velop smart invoice.

Field	Description
Disc./unit	Absolute discount on the unit price in the price unit
Disc.	Absolute discount on the calculated net amount
Disc. %	Discount in % on the calculated net amount after deduction of Disc./unit and Disc.
Disc. 2-5 %	Further discounts in % on the calculated net amount after deduction of Disc./unit , Disc. and Disc. % as well as, if applicable, Disc. 2-4 % .

Note

The discounts are taken into account in the calculation of the net amount. Calculating the net amount requires at least the **Amount** and **Unit price** fields.

The discounts are not taken into account if the net amount is not a calculated value.

The net amount is calculated as follows taking discounts into account:

$$\left(\frac{(\text{Einzelpreis} - \text{RabattproSt.})}{\text{Preiseinheit}} \cdot \text{Menge} - \text{Rabattges.} \right) \cdot \left(1 - \frac{\text{Rabatt\%}}{100} \right) \cdot \left(1 - \frac{\text{Rabatt2\%}}{100} \right) \cdot \dots \cdot \left(1 - \frac{\text{Rabatt5\%}}{100} \right) \\ = \text{Nettobetrag}$$

Example

You have a line item with a quantity of 1,000 pieces. The unit price is EUR 50.00 in the price unit of 100. Your supplier grants you a discount per unit (packing unit) of EUR 3.00. Due to a special offer, you receive a 10% promotional discount. You also receive a 5% loyalty discount.

$$\left(\frac{(50 - 3)}{100} \cdot 1000 \right) = 470,00 \cdot \left(1 - \frac{10}{100} = 0,90 \right) = 423,00 \cdot \left(1 - \frac{5}{100} = 0,95 \right) = 401,85$$

Other factors influencing the net amount

The net amount of a line item in d.velop smart invoice is also calculated in other ways. The calculation of the net amount depends on how you use d.velop smart invoice.

Warning

Please note that the following calculations of the net amount may result in the net amount no longer matching the calculated net amount from the factors **Quantity**, **Unit price**, **Price unit** and **Discounts**.

Entries in the line item field "Gross amount"

If you make an entry for the gross amount of a line item, the net amount is calculated based on the gross amount, provided the calculation is possible with the available information (tax amount or tax code).

If a tax code is available, the net amount is always calculated as follows:

$$\frac{\text{Bruttobetrag}}{1 + \text{Steuersatz}} = \text{Nettobetrag}$$

However, if no tax code is available but a tax amount has been entered, the net amount is calculated using this formula:

$$\text{Bruttobetrag} - \text{Steuerbetrag} = \text{Nettobetrag}$$

Note

In the calculation of the net amount, the tax code takes precedence over the tax amount if the calculation is based on a change to the gross amount.

Following the recalculation, any amount in the line item field **Tax amount** that differs from the calculated tax amount is overwritten with the calculated tax amount.

Entries in the line item fields "Tax code" or "Tax amount"

Note

In the case of changes in the line item fields **Tax code** or **Tax amount**, d.velop smart invoice always attempts to calculate the gross amount.

Only if the net amount, which is necessary for calculating the gross amount, is missing does the calculation of the net amount take place.

If you make entries for the tax code, d.velop smart invoice calculates the net amount as follows:

$$\frac{\text{Bruttobetrag}}{1 + \text{Steuersatz}} = \text{Nettobetrag}$$

Note

Following the recalculation, any amount in the line item field **Tax amount** that differs from the calculated tax amount is overwritten with the calculated tax amount.

If you make entries for the tax amount, d.velop smart invoice calculates the net amount as follows:

$$\text{Bruttobetrag} - \text{Steuerbetrag} = \text{Nettobetrag}$$

Note

If the calculation of the net amount is based on a change to the **Tax amount** field, the tax amount is used for the calculation. The tax code does not automatically take precedence over the tax amount.

This deviates from the applicable calculation rules if the calculation is based on a change to the gross amount.

Calculating the gross amount of a line item

In d.velop smart invoice, the gross amount can be calculated in two ways.

Net amount and tax rate of the tax code used

The gross amount is generally calculated from the net amount of the line item and the tax rate of the tax code used. The gross amount is calculated using the aforementioned values when the net amount is recalculated or entries are made in the **Net amount** field and a tax code is available. The calculation is also performed in this way if a tax code is selected.

$$\text{Nettobetrag} \cdot \left(1 + \frac{\text{Steuersatz}}{100}\right) = \text{Bruttobetrag}$$

Gross amount and net amount

In d.velop smart invoice, you can manually enter the tax amount in a line item. The gross amount is calculated based on the **Net amount** and **Tax amount** values. The calculation with the aforementioned values takes place when the net amount is recalculated or entries are made in the **Net amount** field and no tax code is available but a tax amount exists instead. The calculation is also performed in this way if the tax amount is changed.

Note

You can enter tax amounts that differ from the calculated tax amount. d.velop smart invoice notifies you of this by means of an icon  in the **Tax amount** field and a corresponding message.

$$\text{Nettobetrag} + \text{Steuerbetrag} = \text{Bruttobetrag}$$

Note

In the calculation of the gross amount, the tax code takes precedence over the tax amount if the calculation is based on a change to the net amount.

Following the recalculation, any amount in the line item field **Tax amount** that differs from the calculated tax amount is overwritten with the calculated tax amount.

Note

If the calculation of the gross amount is based on a change to the **Tax amount** field, the tax amount is used for the calculation. The tax code does not automatically take precedence over the tax amount.

This deviates from the applicable calculation rules if the calculation is based on a change to the net amount.

Rounding differences

Rounding differences are normal in incoming invoice processing. These differences arise due to the different methods used in tax calculation (horizontal or vertical method).

d.velop smart invoice uses the horizontal method to calculate the tax amounts. This means that tax amounts and the resulting gross or net amounts are calculated per line item – depending on the basis of the calculation d.velop smart invoice generally rounds the calculated amounts to two decimal places.

The amounts are added up to give the total amount, which is displayed in the totals line below the line item table.

Note

The different methods employed for tax calculation can result in the total amounts in the workflow header data differing from the totals of the line item amounts due to rounding differences.

1.4. Additional information sources and imprint

If you want to deepen your knowledge of d.velop software, visit the d.velop academy digital learning platform at <https://dvelopacademy.keelearning.de/>.

Our E-learning modules let you develop a more in-depth knowledge and specialist expertise at your own speed. A huge number of E-learning modules are free for you to access without registering beforehand.

Visit our Knowledge Base on the d.velop service portal. In the Knowledge Base, you can find all our latest solutions, answers to frequently asked questions and how-to topics for specific tasks. You can find the Knowledge Base at the following address: <https://kb.d-velop.de/>

Find the central imprint at <https://www.d-velop.com/imprint>.