Uptempo

Job Manager and Marketing Data Hub

Administration Manual

Version 8.0

DE
5. May 2025
100600000001

Copyright

Amalienbadstraße 41a, 76227 Karlsruhe (Germany), www.uptempo.io

Your feedback is important to us!

We would be grateful to be notified of any errors you may discover. Just send us an e-mail to documentation@uptempo.io.

Contents

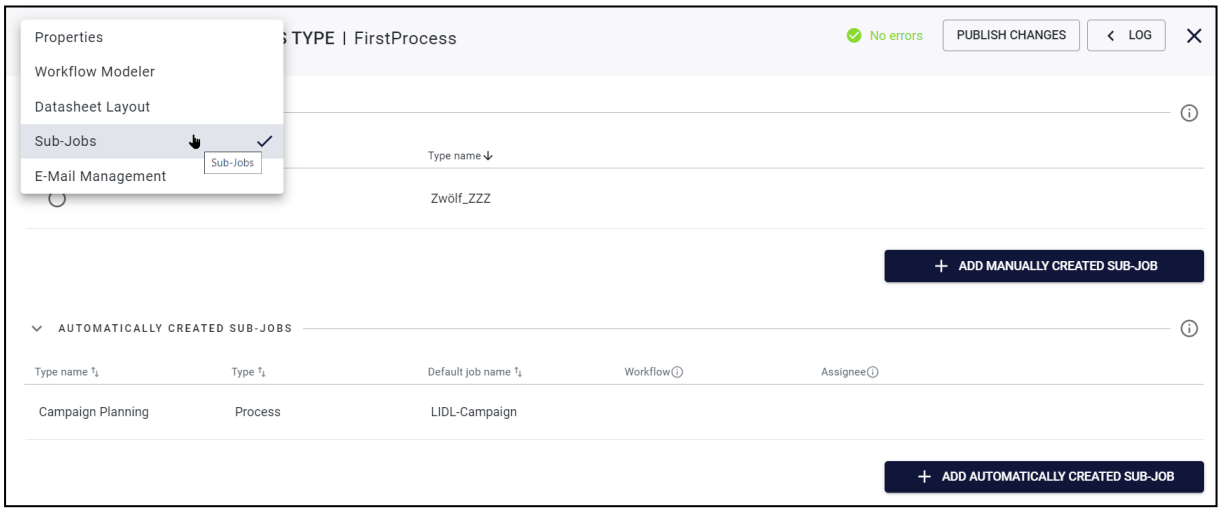
1 New and Changed in This Version

1.1 More Flexible Creation of Sub-Jobs in BPMN Workflows

The functions for creating and managing sub-jobs have been improved: sub-jobs can now be created at any time in the workflow without triggering a status change. This simplifies the modeling of complex workflows. The user benefits from simpler operation and stable system functions when handling complex processes.

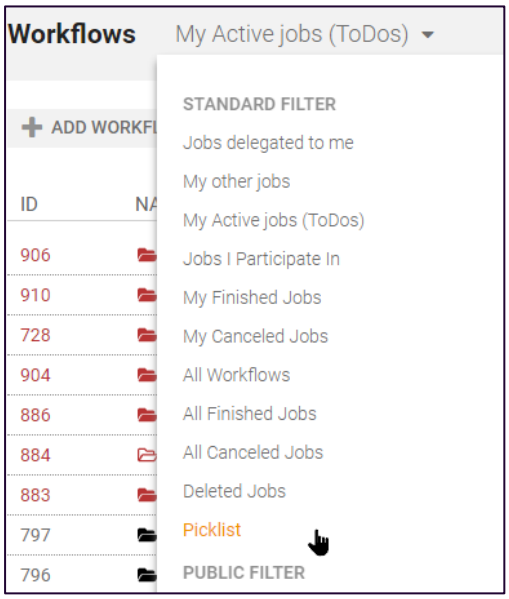
This includes:

- Separate page for sub-jobs in the process data sheet layout
- Validations on deletion to prevent errors when deleting sub-jobs.
- Clear presentation of sub-jobs: Differentiation between job and process sub-jobs in tables.



1.2 New Default Filter for Picklist

A frequently requested feature has now been implemented: From now on, users will see a new default filter *Picklist* in the Job Manager. This lists all jobs that are assigned to the next workflow step without being processed.



The new *Picklist* filter eliminates the need for users to manually create new filters or wait for the Dashboard to appear. Administrators no longer need to create and maintain filters for individual groups. Users can quickly get to the tasks they can work on. This saves valuable time in the daily work with the Job Manager. Since the filter automatically displays only the jobs that are relevant to the respective user, filters no longer need to be adjusted when the group assignment changes.

Overall, the introduction of this new standard filter provides a simpler overview of the selectable jobs. Users and administrators benefit from less effort in filter management.

1.3 Former Assignees Automatically Added as Participants to Jobs

Former assignees are now automatically added as participants in jobs, ensuring access via the "Jobs I Participate In" filter, even without assignee permissions. For full details, see chapter 6.2.

How it Works

- When forwarding a job, former assignees are automatically added as participants.
- The job appears under the "Jobs I Participate In" filter with participant access rights.
- Reassumed assignees revert to assignee status without an extra participant entry.
- Creators remain as creators without a participant entry.
- Participants can be manually removed.
- No email notifications are sent for automatic participant additions.
- Administrators and power users retain full access.

> Example
> An agency assigned in step 3 can open the job in step 7 with participant access rights.

Post-Update Actions

1. Update Access Rights: Restrict participant views to maintain data security.
2. Review Job Filters: Ensure the filter includes jobs where users were previously assignees.
3. Communicate Changes: Inform users about the new functionality and its impact.

Discontinued in Assignee/Participant Context:

- Pre-8: Former assignees were removed from the participants list.
- Post-8.0: They remain as participants for continued access.

> Note
>
> Reverting to pre-8.0 behavior is not possible in versions 8.0and later.

1.4 Terminology Adaptation

The term *iteration* has been replaced by *version* throughout the system. Previously, the revision iterations of jobs were referred to as *Iteration 1*, *Iteration 2*, etc. Now, the correct term *version 1*, *version 2* is used system-wide.
Using the term *version* instead of *iteration* avoids misunderstandings, as the term *versions* has also been used in Review Manager for some time. The user will find consistent terminology throughout the system.

1.5 Inheritance of Variables Based on Technical Names

Previously, variables were inherited based on their displayed names in the user interface. As of Release 7.5, inheritance is based on the technical name of the variable. Basing inheritance on the technical name enhances system consistency and stability by eliminating potential errors caused by using different names for the same variable. Changing the inheritance principle ensures that variables are inherited correctly, regardless of their naming in each language. This supports multilingualism and increases system quality.

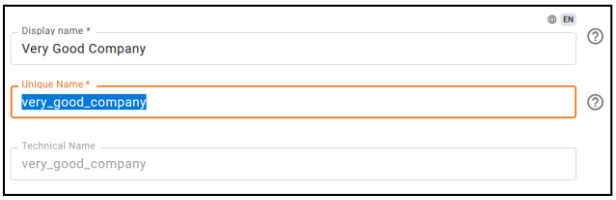
1.6 DSE Administration Improvements

The new features and administration improvements contribute to greater efficiency and user-friendliness of the software. More complex workflows can be modeled more easily and permissions can be controlled more granular. These improvements include the following features:

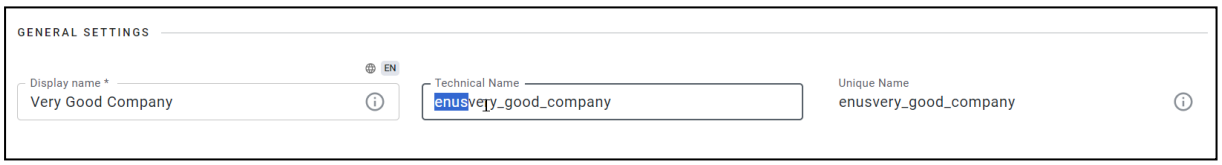
1.7 Use of Technical Names for Types

Previously, the unique name was used in the API and for other reference. This has now been changed. The procedure is now the same as for the variable names on the data sheet.

Before:



Starting with Version 8.0:



With version 8.0 we're introducing descriptive technical names for Jobs. These can be changed if necessary without affecting the database structure. This simplifies the configuration of job types. The technical name is now considered as reference and replaces the unique name in its previous function.

The API has been adapted to call the technical job names. The unique names of the job types remain visible in the DSE administration of an opened type. They are used to generating reports but can no longer be edited.

1.8 Simplified Visibility Settings and Permissions for Process Data Sheets

Version 8.0 simplifies the administration of visibility settings and permissions for process data sheets.

Clear Authorization Control

All roles and their access rights are visible at a glance. The preview now directly shows what the data looks like for different roles. This allows administrators to keep track of everything.

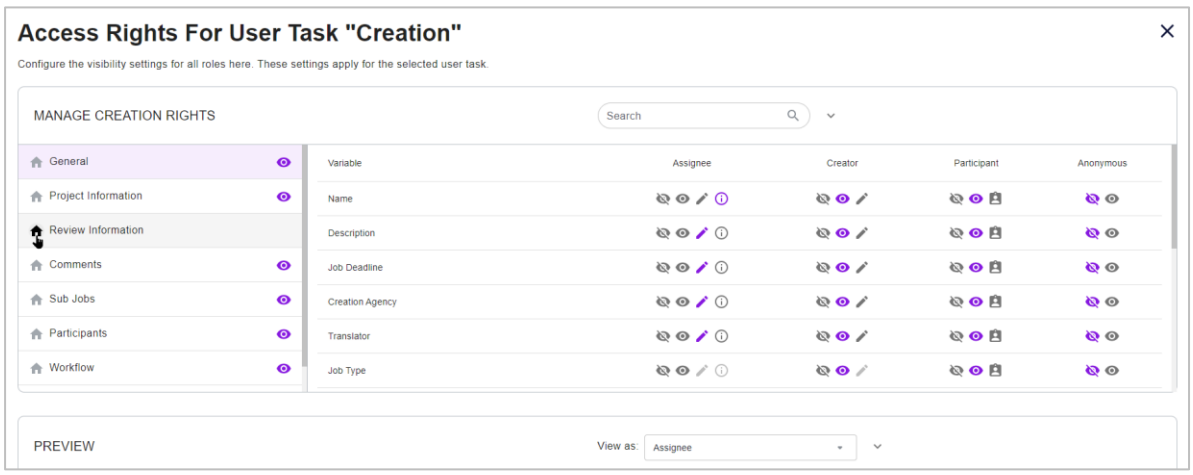
More Flexibility in Process Design

Simpler settings allow for better customization of processes. It's also easier to control data sharing.

Simplified Management of Tabs and Permissions

The administration has been grouped more clearly and is located under the new *Access Rights* button.

The once familiar favorite star icon for the default tab has been replaced with a house icon. Click the house icon of a tab to set that tab as the default when you open the data sheet. A black house icon clearly indicates which tab of a process is displayed by default.



- If a tab is set as default, it can no longer be hidden.
- If a formerly invisible tab becomes the default, it will automatically become visible.

This gives administrators a better overview when defining default views. The visibility of the tabs for individual user groups is still set by clicking on the eye icon. A menu will open with five options. The preview area below shows directly what data is displayed for the selected role.

The revised visibility settings make it easier to adapt and tailor processes. The goal is to configure tab visibility separately for assignees and creators. Tabs should be visible to creators but hidden from assignees. The new CREATOR_ONLY option now hides the tab from assignees with the permission *Act Like Creator*.

1.9 New Dynamic Group Assignment for BPMN User Tasks

The Dynamic Group Assignment option enables the dynamic assignment of user tasks to groups within a workflow based on the scenario. To use this feature, you must create and place the newly introduced *user group* variable on the data sheet. The

variable can be preconfigured with a default entry. If users are required to select a group later, the variable must be set as a required field.

1.10 JS Injection

Use the new *JS Injection* field on the data sheet during type creation or customization. The stored JavaScript code is only executed when the data sheet of this type is opened. Another option is to store a global script. You can store and activate this under *Administration* > *Overview* > *Other Settings* > *Other Settings* under *JS Injection*. a script runs every time the job overview is loaded by navigating to >*Jobs* > *Search*. The advantages are:

- Increased stability and maintainability, as the code is no longer implemented natively via customizations
- Direct access to current product versions, as there are no dependencies on outdated code
- Reduced complexity and simplified visibility by removing redundant and scattered logic

With this change, we're eliminating technical debt and future-proofing the code for future releases. Use only official APIs and documented product features.
See https://developers.brandmaker.com/api/JobManager/7.5/7.5.1.0/.

1.11 Additional Changes and Improvements

In the course of refactoring measures, the integration behavior was standardized and improved. These changes increased user experience through optimized user interfaces and productivity through an accelerated system.

Improved Performance

Under the hood, several measures have been taken to speed up page loading and make the system more responsive. We have made the architecture clearer by cleaning up the code.

Simplified Integration

The integration behavior was standardized and improved in the course of the refactoring measures.

- Increased user experience through optimized user interfaces
- Increased productivity through an accelerated system

1.11.1 Changed Publishing Logic of Job Type Configurations

With this release, the publishing logic for job type configurations has been changed. Previously, parallel changes made by different users could lead to conflicts.

The Most Important Changes

- Unified central database as a single source of truth for configuration data of job types
- Separation of the management of configuration data into administration and runtime removed
- Implemented synchronized sequential process for schedule changes to avoid conflicts
- Categories removed from the publishing process
- Task templates removed from the publishing process
- User settings removed from the publishing process
- Removed DSE module settings from the publishing process

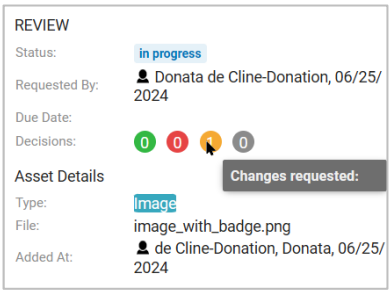
These changes have improved data integrity when managing job type configurations. Parallel work of multiple users and associated problems such as central storage prevent the issue of outdated data. Administrators and users benefit from a consistent view of job type configurations during administration and runtime. Sequential processing of changes to schedules keeps system configuration up to date.

1.11.2 Datasheets in Multiple Languages

Key features now support multiple languages without any issues. The localization of individual variable fields when configuring the datasheet by an administrator has been revised.

Translations are easier to manage and maintain. As a result, the system is better prepared for use in different regions. Users benefit from being able to use the system in their preferred language, and administrators benefit from an easier localization of content.

1.11.3 Additional Enhancements

Feature	Description
Rich Text Editor Enhancements	With the update, the integrated rich text editor has been revised and expanded with many useful functions. The user can work more comfortably and intuitively with formatting options such as fonts, font sizes, and formatting. In addition, the user experience has been improved with new tools such as font and font size selection. This makes it even easier for the user to create and format text. With the update to the new editor version, several minor bugs have also been fixed.
New Permissions	Change Management: To replace the removed administrator permissions for type configuration, we have introduced two new permissions: *Manage Jobs* and *Manage Products*. These permissions provide users with the same level of control and functionality, combining the previous admin rights into a more concise and intuitive set of permissions.
Improved display of the status *Changes requested.* 	When *Request Changes* decision system preferences are enabled, Job Details will no longer display *Approved with changes* but *Changes requested*. This makes the status clearer and easier to understand for the user. **Benefits for users** • Confusion about the actual status of a job is avoided • Uniform terminology in the Job Manager and Review Manager modules • Clarity about open change requests/reviews
Revised access rights control in the Workflow Modeler	The access rights control for the *Workflow Modeler* has been redesigned to make it easier to manage permissions. Users can now be easily granted or denied access to specific workflows and tasks.
Revamped layout management	The datasheet layout has been revised to be cleaner and more user-friendly. This makes creating and customizing layouts more efficient.

1.12 Discontinuations

1.12.1 Removal of SOAP Interface

The SOAP interfaces for Job Manager and Marketing Data Hub have been removed. Previously, certain functions and data could be controlled via SOAP calls. This is no longer possible as of version 8.0. Instead, the REST interfaces must be used.

By unifying on REST, developers only need to focus on one interface technology. This simplifies the development of individual solutions for the platform.

REST is the more modern and widely used technology for API calls. By focusing on REST instead of SOAP, the further development and long-term support of the interfaces is ensured.

The majority of solution providers and platforms today rely on RESTful APIs. The transition will facilitate integration with third-party systems as well as the development of standalone clients.

1.12.2 Removal of MMS

In this release, we've removed the MMS customizations that were previously used to integrate functionality into Job Manager that wasn't covered by the standard product. Such customizations are no longer supported, as the corresponding functions are now available directly in the product (Rest API) or can be implemented via your components.

1.12.3 Permissions

The Job Manager / Data Hub administrator is now enabled to configure all aspects of the module without the need for multiple permissions.

The old set of multiple permissions is now deprecated with the introduction of two new administrator permissions replacing them. See table above.

One of the main changes in this release is the removal of old permissions for job type configuration. The following permissions have been discontinued:

- MANAGE_DEFAULT_TYPES
- MANAGE_OBJECT_NUMBERS
- MANAGE_TYPE_CATEGORIES
- MANAGE_TYPES
- MANAGE_TYPE_CONFIGURATION

- MANAGE_VARIABLES
- MANAGE_VARIABLE_ACCESS_AND_RIGHTS
- MANAGE_DATASHEET_LAYOUT
- PUBLISH_DSE_CHANGES
- MANAGE_TASK_TEMPLATES

Administrators can now access the same feature set at once through the permissions Manage Jobs and Manage Products, which fully replace the deprecated ones. These two permissions allow you to configure the respective modules. When using the *Data Hub*, both permissions must be active.

2 Useful Information for Getting Started

This guide helps administrators set up the Job Manager and Data Hub modules. It provides information to help you configure and manage these modules effectively. The content is intended for experienced administrators and includes

- An overview of an administrator's responsibilities
- Detailed instructions, tips, and information about each feature and setting for the *Job Manager* and *Data Hub* modules.

Jobs, Processes, and Data Objects

Jobs and processes are managed in the *Job Manager* module. Jobs are combined with a classic workflow, processes with a BPMN workflow. Jobs and processes represent standardized workflows in your company.
In the *Data Hub* module, users manage data objects, which are always combined with a classic workflow. In data objects, a standardized workflow captures data that can be used, for example, as a data source for other modules.

Classic Workflows and Tasks

One or more classic *workflows* are assigned to each job and each data object. A classic workflow consists of several *workflow steps*. The administrator assigns a user group to each step. When forwarding, the user selects a responsible person from this group. In addition, *tasks* can be added to each workflow step so that typical activities are created automatically when the job is created.

BPMN Workflows

BPMN stands for **B**usiness **P**rocess **M**odel and **N**otation and is used for the graphical representation of business processes. A process is always combined with a BPMN workflow, and the workflow is represented in BPMN notation.

Custom Objects and Structures

By using custom objects and structures, you can, for example, create text modules that can be reused in the *Brand Templates* module for editing documents. In addition, the values for a dropdown list can be loaded from a custom structure.

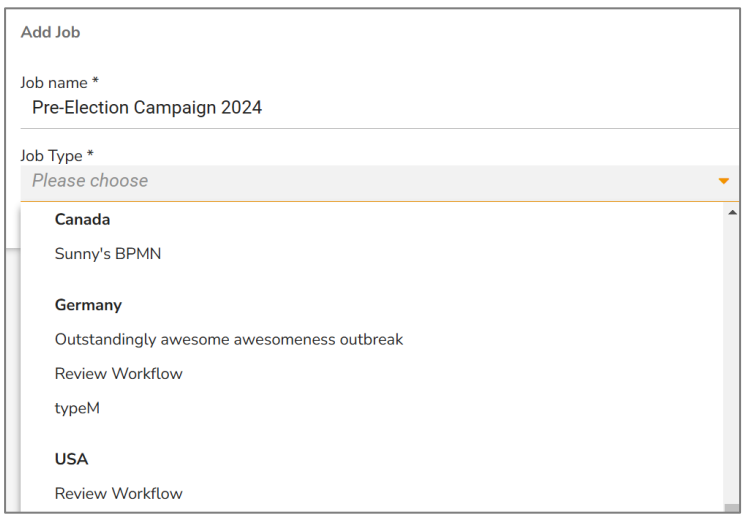
Variables

Different types of variables are used to collect and output data and information. The variables are placed on a datasheet using drag and drop. There are two different types of variables:

- Variables created automatically that allow you to use the basic functions, for example, for displaying the creator or the current workflow step.
- Variables that can be created individually (text fields, selection fields, or an asset selector, for example) can be created and configured as required.

Categories

To organize job types, you can add your categories in *Administration* > *Overview* > *Datasheet Engine* > *Other Settings* > *Categories*. When creating or editing a job type or data object type, the administrator selects one or more categories from the drop-down list in the Categories field to which this type is assigned. Later, when users create a new job, the Job Types drop-down list is uniquely organized according to the assigned categories. In the following example, the same job type *Review Workflow* is available for selection once under the category *USA* and once under the category *Germany*.



Jobs/Processes and Sub-Jobs/Processes

A job or process collects all the information required to complete a task, such as creating a new campaign. Additional tasks can be mapped to sub-jobs or sub-

processes. Sub-jobs or sub-processes can use an independent workflow. They allow you to structure workflows and make dependencies between them visible.

> Note
>
> It is possible to perform synchronization between jobs or processes of a certain type and corresponding planning elements in the module Marketing Planner. This synchronization is set up by Uptempo.
>
> If you require any further assistance, please speak to your Uptempo contact person.

Products and Sub Data Objects

A *data object* groups together all the information related to a data object. If the data object consists of multiple parts, they can be represented through *sub data objects*. A sub-data object can use a different workflow independently of the data object. The sub-data objects allow a precise mapping of the data structure of the data object.

User

The access to and visibility of tabs and the field functions (variables) placed on them can be restricted when you configure the type for each workflow step. For this, users are divided up according to *Assignee/Processor*, *Creator*, *Other participants* and *Anonymous*. The user role to which you belong determines the datasheet tabs and variables that you can view or edit.

User	Description
Creator	You are the creator (owner) of a job, process, or data object if you have created the job or data object. The creator can be changed at a later stage. The creator can only be changed subsequently for the types *job* and *data object*.
Assignee/Possessor	You are the (current) editor/assignee of a job, process, or data object if you are responsible for the current workflow step. You can also be a member of an assigned group, where each person has the same rights, but you choose the workflow step to be edited, enter the data and forward it to the next step.
Participant	You are a participant if you are invited to the job, process, or data object as a participant. As a participant, you are not responsible for the current workflow step, but you can follow the progress of the job, process, or data object. Participants can be added to a discussion.

User	Description
Anonymous	Anonymous users are any users who are not the creator, assignee/processor, or participant, but who can still access a job, process, or data object. In general, anonymous users only have read access to variables. By default, anonymous users do not have access to the variables. These remain hidden for them. However, administrators can set up read access.

Note

You cannot subsequently modify the creator of a process type (BPMN workflow).

2.1 Action Required After an Upgrade

> Note
>
> You can skip this section for an entirely new installation, and when upgrading from 7.3 or higher to the latest version.

Reason for Taking Action

You have just upgraded from a 7.2 or earlier system to the latest version. After upgrading a system that was set up before version 7.3, it is necessary to migrate the types. This step ensures that the types work correctly. The migration is usually done by Uptempo.

The reason for the migration is the significant change in the data structure after the upgrade.

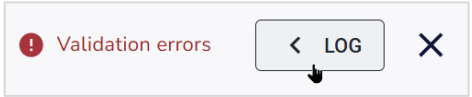
Required Steps

Only authorized users can perform the migration. To achieve this, the user must have a custom role in the Administration with the *Manage Types* and *Migrate Types* permissions.

Cleaning up Old Data

1. After the upgrade, you should manually repair non-validated types or delete them on a case-by-case basis if they are no longer needed.
2. Go to > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance*.

   You are on the *Publish Changes tab*.
3. If you don't see any changes in the *Publish Changes tab,* go directly to step 12.
4. In the list of changes, identify all types with a red mark and the note, *Please check type for errors*.
5. Click on the linked name of the type below the note text to open the datasheet of the faulty type.
6. Click *Log* in the upper-right corner to view the result of the validation.

   The change log and validation result are displayed on the right.

You now have these options for the changes:

- The Discard *Changes* and *Publish Changes* buttons are both available. Just close the job type.
- You can only select the *Discard Changes* button. Click *Discard changes,* and then *click Confirm*.
- There is no defined workflow for the job type. Just close the job type.

7. If you choose not to correct the errors individually, click the *Discard changes* button at the bottom of the changelog.
8. *Click Confirm*.

   The window is closed. The type is reset to the state before the validation failures.
9. Optional: Use the ☰ drop-down menu to switch to the corresponding subpage (*Workflow Modeler* or *Workflows*) to immediately fix errors in the BPMN workflow or a missing workflow.
10. Click *Publish changes* in the datasheet.
11. Go back to the > *Administration page* > *Overview* > *Datasheet Engine* > *Maintenance* and repeat steps 7-8 for all available job types until no more errors are displayed.
12. After each type validation is complete, go to > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance.*
13. On the *Publish Changes* tab, all pending changes must be marked with a green checkmark.
14. If this applies to all pending changes, click *Publish*.
15. Click *Confirm*.

The data has been repaired.

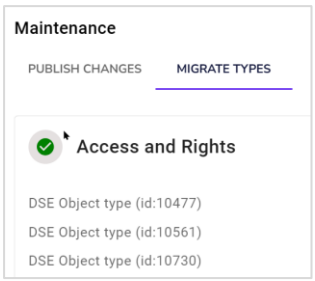
Migration

1. Switch to the *Migrate Types* tab when the *Publish Changes* tab shows no more pending changes.
2. Run > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance* > *Migrate Types*.
3. This service works in the background to adapt any existing type and publish it in the new format that is valid.

You have completed the migration. Following these instructions will ensure that the types are migrated correctly and that the system functions smoothly after the

upgrade. After successful migration, you can open and create jobs, processes, and data objects as usual.

Migration can only be performed once. This status message indicates that a migration has already been performed and the button *Migrate* is deactivated.



Typical Problems When Migration is not Performed

If the Job Manager module cannot be used after the upgrade and no new types can be created under > *Administration* > *Overview* > *Datasheet Engine*, the migration of the types may not have been carried out yet.

If you have any further questions, please contact your Uptempo contact or support.

2.2 Display, Technical, and Unique Name

Each type and each custom variable have three different names:

- Display name *
- Technical Name
- Unique Name

Property	Display Name	Technical Name	Unique Name
Usage	The *display name* is the name that is used to show the type or variable on the interface and that is used for the inheritance. You can create the display name in various languages.	The *technical name* is used to operate objects via interfaces of external systems (for example, via REST). The technical name is also used when grouping variables and is used for inheritance.	Must be used when creating formulas in Formula type variables. The unique names of the job types remain visible but cannot be edited. They are still used for report generation. The APIs do not access the *unique name*.
Input	You enter the *display name* when you create a type or variable.	The *technical name* and the *unique name* are derived and generated from the *display name* when a type or custom variable is created. Changing the inheritance principle ensures that variables are inherited	

Property	Display Name	Technical Name	Unique Name
		correctly, regardless of how they are named in different languages.	
Change	The *display name* can be changed and edited in any way.	The technical name can be changed.	The unique name cannot be changed.
Restrictions	There are no restrictions for the display name. However, long names may not be displayed in full under certain circumstances.	The characters [a-z], [0-9], and [_] are used for the name.	
Derivation	—	• If the display name begins with a number, the name "type_" is prefixed to the technical name. • Space characters are replaced with an underscore "_". • Umlauts and special characters are removed. • Uppercase letters are replaced by lowercase letters. • Restricted to a maximum of 255 characters.	• Like technical name; additionally restricted to a maximum of 24 characters.
Uniqueness	The display name can be used multiple times for each type.	Both the technical name and the unique name must be unique for each type. If multiple variables of the same type with the same display name are entered, consecutive numbering is added during the derivation (for example, price_1, price_2).	

Note

Up to version 7.4, it was possible to edit unique names. However, as of version 7.5, both the technical job name and the technical variable name are referenced by adapting the API interfaces. Therefore, in the open job type, only the technical and display names of the type and variables can be edited during creation and editing. The unique name is locked for editing.

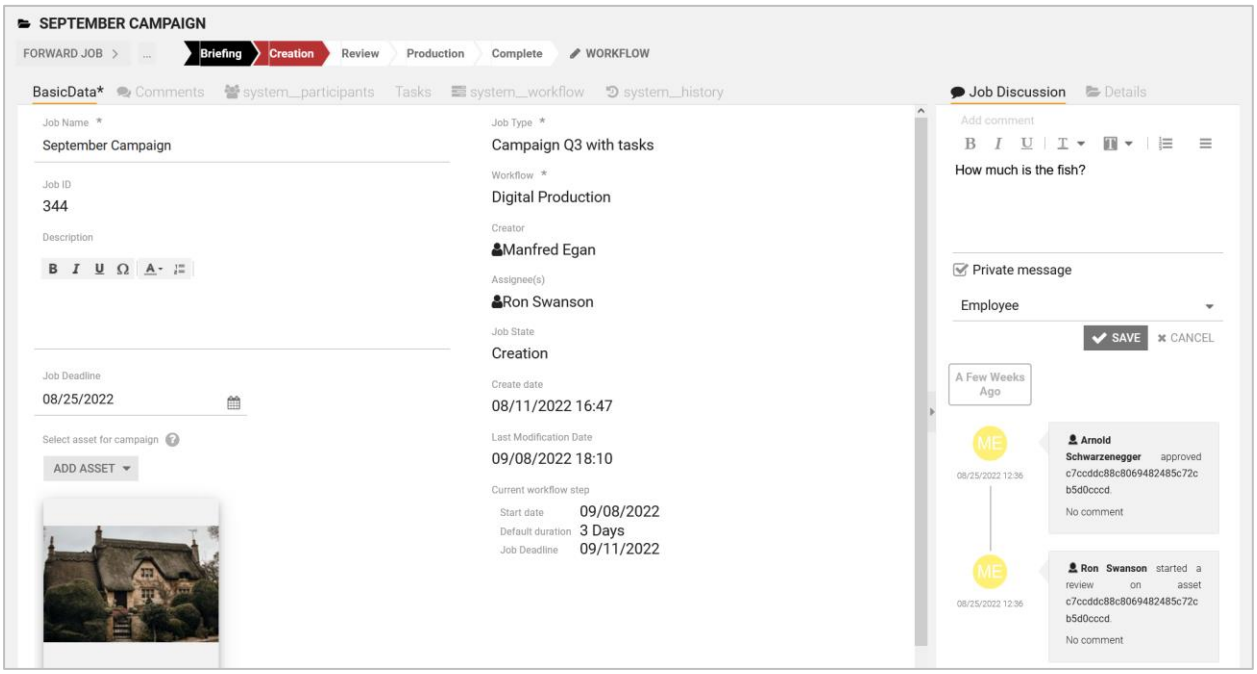
2.3 Datasheet

All the information about a job or data object is collected and mapped on a *datasheet*. You can group together the different variables clearly on various tabs and place them easily using drag and drop.

Among others, the following information is displayed on the datasheet:

- The users who are involved/participating in the job or data object
- The current state (the current workflow step)
- The creator and the current assignee
- The predefined project workflow for the job or data object, including any tasks that have been defined.
- The relevant information for the job or data object

> Note
>
> The appearance of the datasheet can be defined to suit the purposes of individual customers. Among others, the appearance is determined by the number of tabs created, their names, and the field functions that are used on them. The *Basic Data* and *Comments* tab are created automatically; they can be renamed but *cannot* be removed. You can show and hide tabs for specific users individually for each datasheet. See Chapters 3.5.2 and 4.7 for setting visibility and editability.

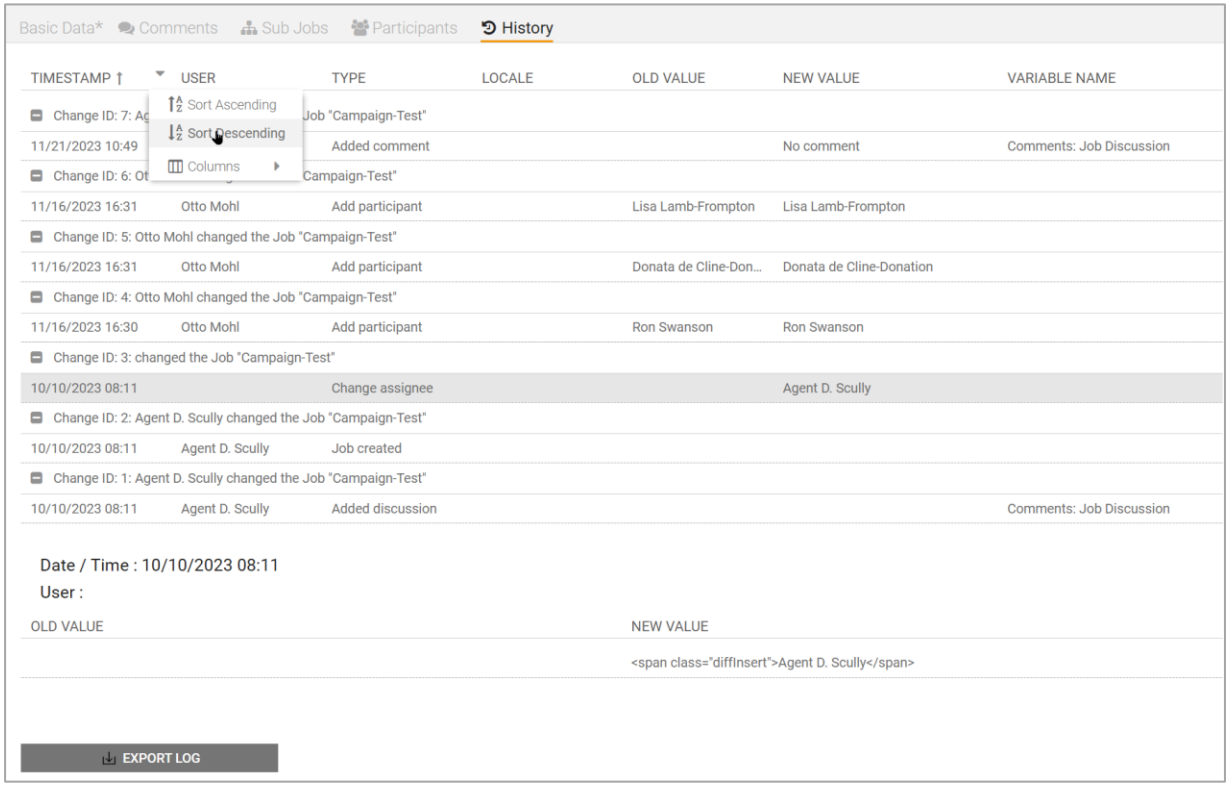


2.3.1 Change history

On the open datasheet, navigate to the *History* tab to open the change history. You can use the change history to monitor when an object was processed or edited. It also records the amount of time it took. Navigate to the *History* tab in the currently open datasheet to open its change history. The following information is displayed in a table overview:

- Change ID: Consecutive numbering of the changes (only with Job Manager)
- Timestamp: The time at which the change was made
- User: The name of the user who added or edited a comment or discussion
- Type: The type of editing or change (for example, the addition of a new comment)
- Locale: This column is not relevant in Job Manager.
- Old value: The original value of the edited variable
- New value: The new value of the edited variable
- Variable name: The name of the variable or discussion that was edited. Select the line to display the new and old value in the lower area of the dialog window.

Click *Export log* to export the change history as an XLSX file.

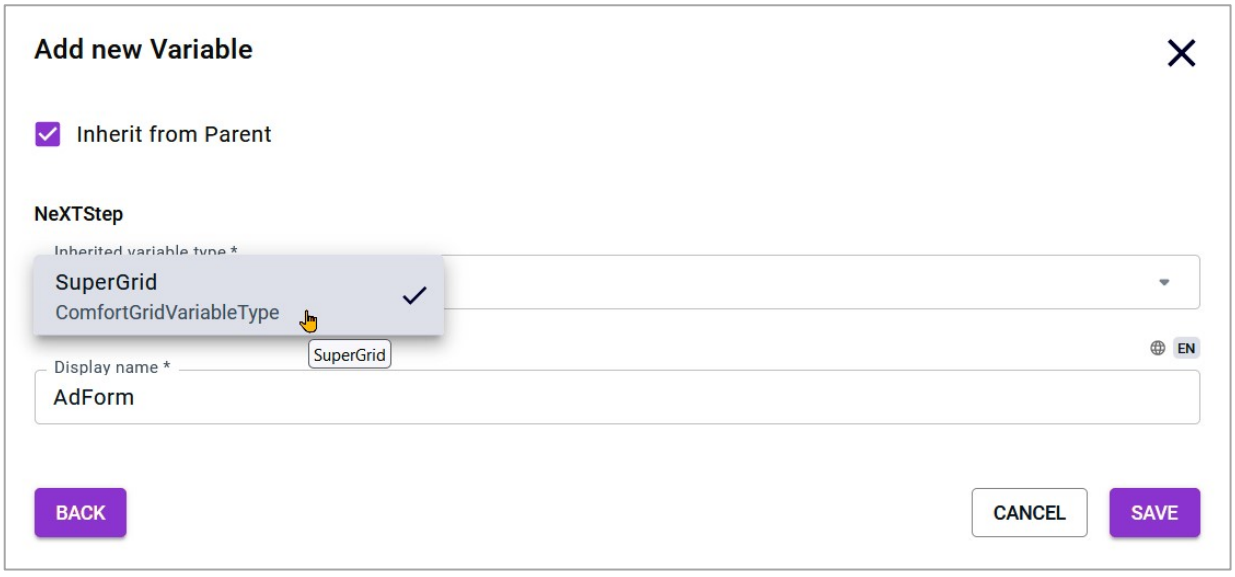


2.4 Inheritance for sub jobs and sub data objects

Sub-jobs and sub data objects can inherit values from a parent job or data object. If the values in the parent job or data object change (for example, a deadline), the value is also updated in the sub-job or sub data object. In the subtype, you can configure the following options using the *Inheritance* selection box:

Note: Note that the inheritance can be created only when you create a new sub-job type or sub data object type. The *Inheritance* field cannot be changed at a later stage!

- No inheritance: In this case, variables of the subtype do not inherit a value from a parent job or data object.
- Single parent: The variables can inherit; however, only one single type is permitted as the parent type. Define the parent types from which the sub-job or sub-data object will inherit. In this case, you have the option of specifying the parent type variable to inherit directly when creating the variables (see screenshot).



- *Multiple parents*: Select this option if the parent types from which a subtype inherits cannot be clearly defined. For example, the *Translation* sub-job can be permitted for both the parent job types *Brochure*, *Manual*, and *Flyer*. Activate the option *Inherit from parent* in the subtype for the inheriting variable. In this case, the link is established via the display name, which must be identical for the variable in the parent type and subtype.
  Users can click on the inheriting variable on the datasheet to break the inheritance.
- In this case, the last inherited value remains in the sub-job or sub data object, even if the value is changed in the parent job or data object. By clicking , users can restore the inheritance.

2.5 Localization

> Note
>
> It is worth noting that you will only be able to use this feature in the Data Hub.

You use the *Localization* function to adapt data objects to the conditions of regional markets in the Data Hub. A localized data object contains a separate datasheet that is adjusted to local conditions and that uses an ID for each locale.

You edit each datasheet in a separate workflow and can select different workflow types for each localized datasheet. This is significant, for instance, if there are different legal requirements in the territories and a legal check or certification is required.

You can use different workflow types to define different rights and visabilities for the required fields for each locale.

To enter values that are identical for a data object in each locale just once, flag the variable as language-neutral. Examples are international data object names or technical data in standardized measurement systems (for instance, measurements in the metric system, weight specifications in kilograms).

You can also create sub data objects for localized data objects that inherit their values. Note that inheritance is only possible between localized parent data objects and localized sub data objects: The sub data object variant for *Spain* inherits its values from the parent data object variant *Spain*.

2.5.1 Configuring localization

Follow the process below to configure the localization. Follow all the required steps for setting up the Data Hub module (see Required Work Steps, page 38):

1. Create the required locales (see Locales, page 34).
2. Define the required workflows. For more information, see the configuration manual.
3. Create a data object type with the Enable Localization checkbox activated (see Types, chapter 3.3).
4. Note: The Localization field can only be edited when you create a type.
5. Assign all workflows required for the localized data object to the type (for Type configuration, see chapter 3.4.2).
6. Create the variables that are used for this data object type.

7. For variables that are identical for each locale, activate the Shared value checkbox (for variable descriptions, see chapter 5).
8. Publish the changes (see Publishing changes on page 78).

2.5.2 Locales

You can use the locale to describe the territories that are relevant to you using a small amount of data. You can enter the following values:

- Name: The unique ID of the locale.
  Warning! Entering other characters as digits leads to errors! Use consecutive numbering that continues the numbering from the last created territory.
- Display name *: Include the names that you want to use in the various languages for the locale. Use the following structure: ~{language code}nameserver multiple names one after the other. Note that you only have to enter the translations for the interface languages of your Uptempo system.
  - *Language code*: language code according to ISO 639-1 in uppercase (for example, EN for English, DE for German).
  - *Name*: name of the locale that is displayed in the selection list in the datasheet of the localized data object.
  - Example: The *Spain* entry in the languages English, German, and French:
    ~{EN}Spain~{DE}Spanien~{FR}Espagne
- Attributes:

Name	Function	Edit
uniqueName	Unique name, which must be unique within the locale	Mandatory field
scriptCode	The character set used for print characters (for example, Cyrillic or Simplified Chinese)	—
numbers	The character set used for numbers	—
languageCode	Language code according to ISO 639-1 in lowercase	Mandatory field
image	Load an image to represent the territory visually. If the field is empty, the flag for the selected country code is displayed if it is stored by default.	Optional

Name	Function	Edit
default	Standard locale setting: true = the locale is the standard locale. false = the locale is not the standard locale. Warning! Only one locale must be flagged as the default locale at all times.	—
currency	Currency entry for the locale	—
countryCode	Enter the country code of the territory	Mandatory field
collationParameter, collation	Parameter for setting the character sorting in the relevant language	—
calendar	Calendar form, such as the Gregorian calendar	—

2.5.2.1 Configuring or Changing the Locale

1. Choose > Administration > Overview > Data Structures & Workflows > Custom Objects & Structures > Custom Objects.

4. In the *Select custom structure* selection list, select the structure `L10N_Locale(L10N_Local)`.
5. Click *Select*.
6. To create a new locale: In the *Create new custom object* field, enter the name of the new locale.
7. Click *Create*.
   OR
8. To change an existing locale: In the *Choose available custom object* selection list, select an existing locale.
9. The following fields are always mandatory:
   - Name
     Warning! Use consecutive numbering that continues the numbering from the last created territory.
   - Display name *
   - Attribute uniqueName
   - Attribute language code
   - Attribute country code
10. Optional: edit the following fields:
    - Attribute default for the locale that is the standard locale.

- Optional: Attribute image

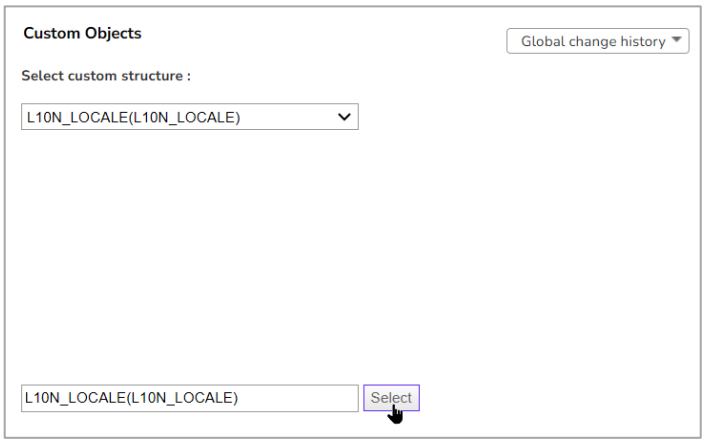
11. Click *the Save* button at the end of the attribute list.

You have configured the locale. Repeat the process if additional locales are required.

2.5.2.2 *Deleting a Locale*

> Warning! Data loss!
>
> Do not delete any locales that are being used in *the Brand Template Builder* module.

1. Choose > Administration > Overview > Data Structures & Workflows > Custom Objects & Structures > Custom Objects.
2. In the Select custom structure selection list, select the structure L10N_Locale(L10N_Locale).
3. Choose Select button.



4. In the Select available custom object selection list, select an existing locale.
5. Click the Delete button below the attributes.

You have deleted the locale.

2.6 Synchronization

2.6.1 Synchronization with Marketing Planner

All users of the Marketing Planner module can use this feature. Basically, it is possible to set up a job type so that planning elements are created, or data is synchronized in the Marketing Planner when the job is edited. For this purpose, the datasheet of the job type must be configured accordingly so that the position in the element tree as well as basic data is mapped correctly. In addition, for the combination of job type and workflow it is defined in which workflow steps the data is updated in the Marketing Planner when saving the job. You can also define whether the planning element is deleted if the job is canceled or deleted.

> Note
>
> This synchronization is set up by Uptempo. Please don't hesitate to contact your service manager for further information.

2.6.2 Synchronization with *Activities*

This feature is available to users of Plan *& Spend* > *Activities*. Which fields of a BPMN workflow are synchronized with the *activity* is defined in the settings of the *activity*.

In *the Workflow Modeler,* send tasks must be configured for this purpose. See chapter 4.2.5.1.

3 Required Work Steps

You must perform a range of work steps for the configuration. It is worth mentioning that the process differs based on whether you are using a conventional workflow created with the previous function or a BPMN workflow.

3.1 Jobs and Data Objects

Prerequisite

- You have created users and a user group.
- You have created a workflow.

Note the detailed descriptions for users, user groups, and workflows in the configuration manual.

Step by step

The following work steps are required or can be carried out as options for the configuration of Job Manager and Data Hub:

1. Create a new type and assign a module (see chapter 3.3).
2. Link the types with one or more workflows (see chapter 3.3.4).
3. Design datasheets for a type (see chapter 3.5).
4. Create and define variables (see chapter 3.6).
5. Optional: Define conditions for displaying a variable and access rights for variables for each workflow step (see chapter 3.6.1).
6. Optional: Define variables for the e-mail notification (see chapter 0).
7. Optional: Create a category (see 3.8.1).
8. Optional: Configure the format of the unique object numbers (see chapter 3.8.2.1).
9. Optional: Define the default type for new jobs and data objects (see chapter 3.8.2.1).
10. Optional: Configure the settings for the *Only Briefing* type and theme navigation (see chapter 3.8.3).
11. Publish changes (see chapter 0).

3.2 Processes

Prerequisite

- You have created users and a user group.
- You have created the custom structures required for your decisions.

Note the detailed descriptions for users, user groups, and workflows in the configuration manual.

Step by step

1. Create new types and assign a module (see chapter 3.3.2.1).
2. Design the datasheets for a type (see Designing the datasheet layout in chapter 3.5).
3. Create and define variables (see Creating and managing variables in chapter 3.6).
4. Optional: Define conditions for displaying a variable and access rights for variables for each workflow step (see Rights and visibilities of variables in chapter 0).
5. Create a BPMN workflow for the type (see chapter 4).
6. Optional: Define variables for the e-mail notification (see E-mail notification in chapter 0).
7. Optional: Create categories (see Type categories in chapter 3.8.1).
8. Optional: Configure the format of the unique object numbers (see Object number configuration in chapter 3.8.2.2).
9. Optional: Define the default type for new jobs and data objects (see Default types in chapter 3.8.2.1).
10. Optional: Configure the settings for the Only Briefing type and theme navigation (see Settings in chapter 3.8.3).
11. Validate the created type (see chapter 3.9.2).
12. Publish changes (see Publish changes in chapter 0).

3.3 Types

The type is the central element of the datasheet Engine. The type determines which basic data is created for a job, a process, or a data object. You create types under > *Administration* > *Overview* > *Datasheet Engine* > *Types*.

Prerequisite

To reach that page, it is required that the *Manage Jobs* permission is assigned to your role.

If you open a type for editing or create a new one, the Properties page opens first. Instead of tabs, a ☰ dropdown menu is used to switch to the other pages with their respective editors.

- Workflow Modeler, see chapter 4.1.1.
- Datasheet-Layout, see chapter 3.5.
- Sub-Jobs, see chapter 3.3.3.
- E-Mail Management, see chapter 4.4.

3.3.1 Properties

The following table shows the properties of the job and data object types. A process has the same properties as a job except for the inheritance function. Note that you set the *Name* (and thus the displayed name), *Type*, *Inheritance,* and for Data Objects, the *Localization* property when you create them. Type and inheritance cannot be changed after their creation.

Name	Description
(Displayed) name	Define the name that is visible to users. You can create the displayed name in different language versions. See chapter 2.2.
Type	The type determines which basic data is created for a job, a process, or a data object.
Inheritance	Note: that inheritance can only be defined while you create a new type. The *Inheritance* property cannot be changed afterward! The function can exclusively be used for jobs and data objects. Configure the inheritance: • *No inheritance*: No datasheet variable inherits values from a parent datasheet. • *Single parent datasheet*: Only one job type is available as a possible parent datasheet. • *Multiple parents datasheet*: Multiple job types are available as possible parent datasheets. Refer also to chapter 2.4.
Localization	Note: Can only be activated when creating a new data object type. Select the checkbox at *Enable localization* if you want to create localized variants of a data object. See Localization, chapter 2.5.
Unique name, technical name	Note: Only accessible in the editing dialog box. See chapter 2.2.
Description	Note: Only accessible in the editing dialog box. Enter additional information about the type that is displayed when it is created.
Categories	Note: Only accessible in the editing dialog box. Select the categories to which the type is assigned. When the item is being created, the types are displayed in categories.

Name	Description
Type can only be selected by	Note: Only accessible in the editing dialog box. Specify which organizational unit, user group or VDB group can select the types. Note: Note that this setting does not affect the visibility of jobs based on this type.
Activating an access control.	Note: Only accessible in the editing dialog box. If you activate this checkbox, only users who belong to the same organizational unit or an organizational unit in the hierarchy below it, such as the creator and processor of the job or data object can view the jobs derived from this type.
Parent datasheets	Note: Only visible in the editing dialog box if *Inheritance* = *single parent datasheet* was specified when created. Select a type.
Jobs/processes and data object type only exist as sub-job/process/data object	Activate the checkbox if the job, process, or data object may only be used as a sub-job, sub-process, or sub-data object.
Manually adding sub-jobs/sub-data objects	Note: Effective for Job and Data object types. Only accessible in the editing dialog box. Activate the checkbox if the user is allowed to add sub-jobs or sub-data objects to the job or data object manually.
Permitted sub-job types/data object types	Note: Only accessible in the editing dialog box. Only for Job and Data object types. The *Manual addition of sub-jobs* checkbox must be selected. Specify which types can be added as sub-jobs/data objects during creation.
Selected sub-job/data object type	Note: Only accessible in the editing dialog box. Only for Job and Data object types. The *Manual addition of sub-jobs* checkbox must be selected. Specify which sub-job or sub-data object is added by default during creation.

Name	Description
Automatically added sub-jobs/data objects	Note: Only for job and data object types. Only accessible in the editing dialog box. Specify which sub--jobs or data objects are added automatically when the item is created. Click the plus sign to configure a sub-job or sub-data object. Define the following settings: • *Job/Data object type*: Specify the sub-object type. • *Workflow*: Defines the workflow for the sub-object. • *Processor*: Defines which user is assigned to the sub-object. Depending on the settings of the selected type, further selection fields may open up • *Default job name*: Set a default job name. • *Optional sub-job/data object*: If you activate the checkbox, the user can choose whether the sub-object is created. If the checkbox is deactivated, the sub-object will always be created automatically.
Use for synchronization	Note: Only accessible in the editing dialog box for process types. Once synchronization is enabled, the function cannot be disabled. Activate the switch if the type is used for synchronization of a process with a planning element in the module *Marketing Planner*. If you have further questions about synchronization, please contact your Uptempo contact person.

Note

The "Job Deadline" provided in the parent job is inherited to the sub-job as the default value.

3.3.2 Manage Types

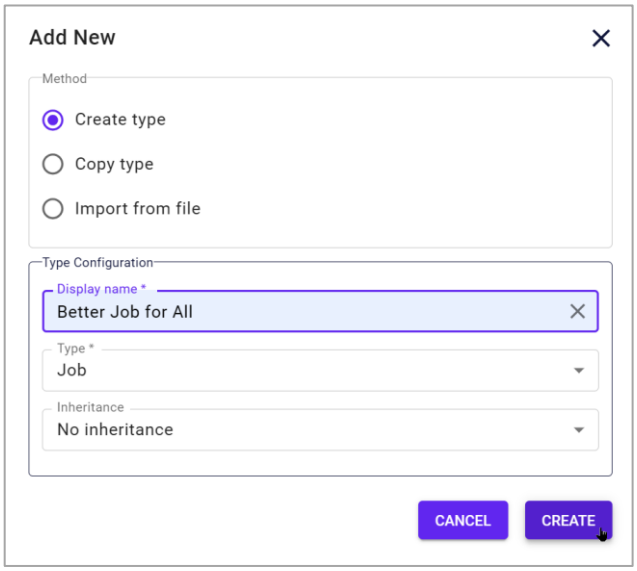
The chapters below describe how to manage types. This includes the following tasks:

- Create Type, see Chapter 3.3.2.1
- Edit Typ, see Chapter 3.3.2.2
- Copy Type, see Chapter 3.3.2.3
- Localize Display Name, see Chapter 3.3.2.4
- Delete Type, see Chapter 3.3.2.5

3.3.2.1 Create Type

1. Choose > *Administration* > *Overview* > *Datasheet Engine* > *Types*.
2. Click the button *Create*.

   The following creation wizard is displayed, where *Create type* is already preselected:



3. Enter a name in the *Display name* field.
4. Select which type you want to create: Job, Data Object or Process.
5. For job and data-object types: Specify the inheritance. Note that you cannot change this setting at a later time.
6. For data object types: Select the checkbox if you want to create localized variants of a data object. Note that you cannot change this setting at a later time.
7. Click *Create*.

When you create a new type or open an existing type, the *Properties* page always opens first.

8. Edit or add to the type properties. See chapter 3.3.1.
9. Edit or create the workflows. See chapter 3.4.
10. Optional: Use the *Log* button to display the sidebar with the change log and validation result.
11. Click the *Publish Changes* button.

    The *Publish Changes* confirmation dialog opens.

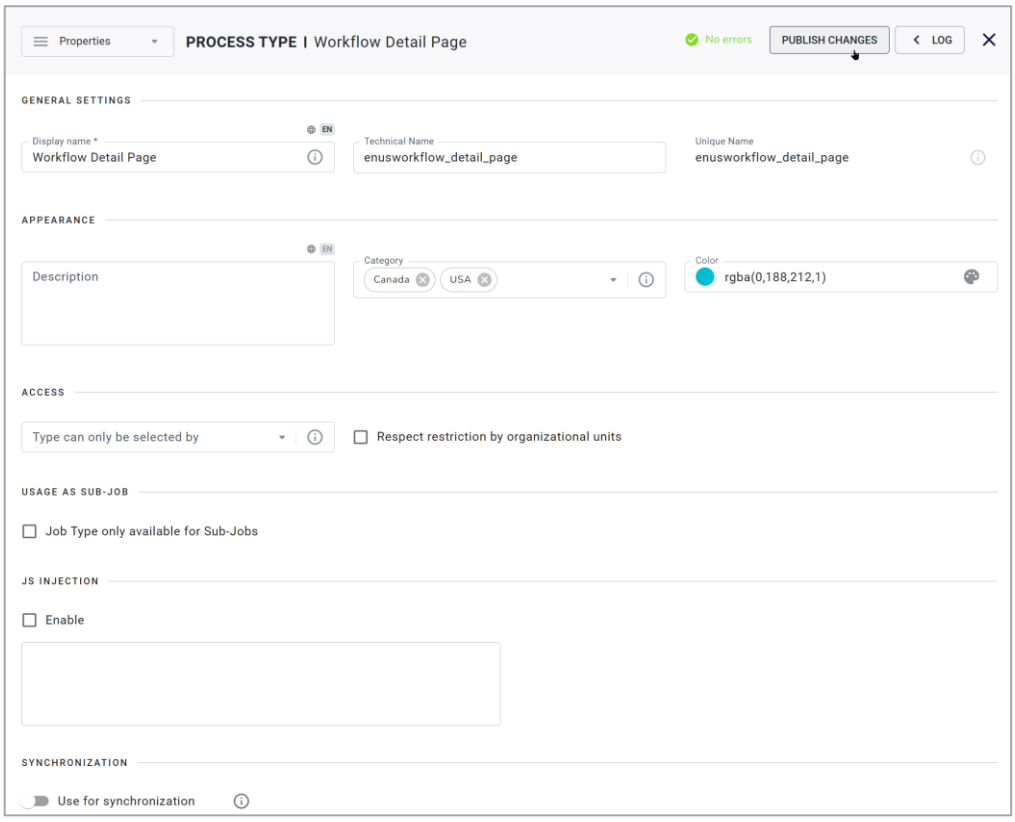
12. Click *Confirm*.
13. Close the page by clicking on the close box ✕.

You have created the type *and* published the changes.

3.3.2.2 *Edit Type*

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > Types.
2. In the table, click the ✎-icon at the end of the row for the type whose properties you want to edit.

   The Job type *Properties* page is displayed:

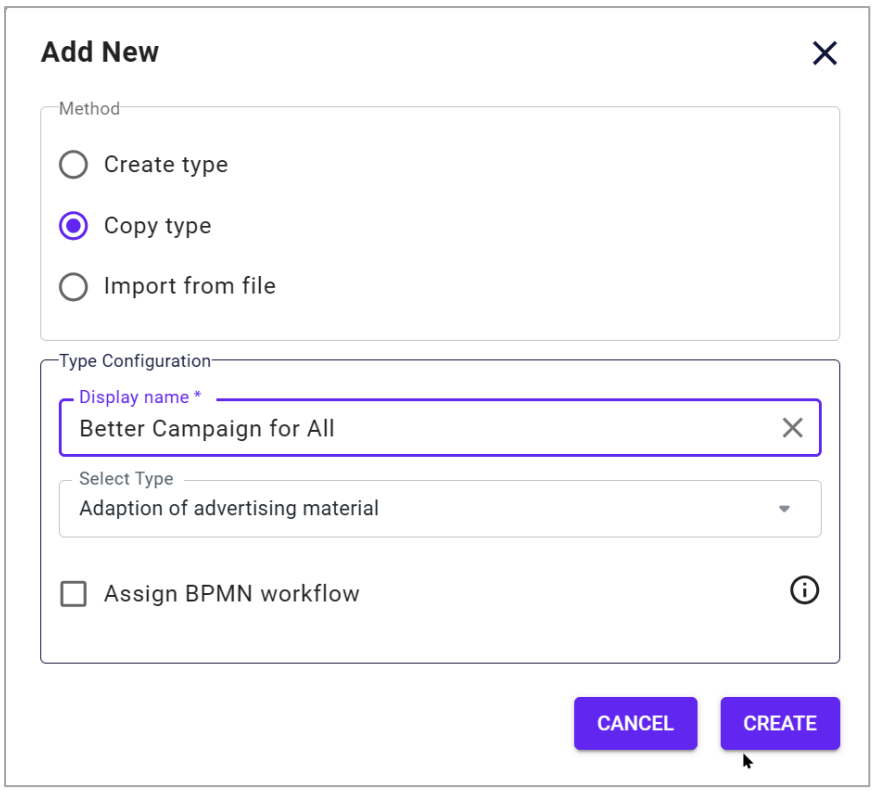


3. Edit or add to the type's properties. Please refer to Chapter 3.3.1.
4. Changes are saved directly, but not published, i.e., they are not yet effective.
5. Optional: Use the *Log* button at the top right to display the sidebar with the change log and validation result.
6. Click the *Publish changes* button.
7. The *Publish changes* confirmation dialog opens.
8. Click *Confirm*.
9. Close the page by clicking on the close box ×.

You have edited the type and published the changes.

3.3.2.3 Copy Type

If you copy a job type with classic workflow, the option Assign BPMN-Workflow is available. If you have activated the checkbox, the classic workflow will be discarded, and you can then create a new workflow for this type in the BPMN Workflow Modeler.



1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Types*.
2. Click the *Create* button.
   The *Add new* dialog opens.
3. Select the *Copy type* method.

4. Enter a name for the copy in the *Display name* field.
5. In the *Select Type* menu, search for and select the type to be used as a template for the copy.
6. Optional: When copying a classic job, you can decide to switch to BPMN workflows.
    a. To do this, select the *Assign BPMN workflow* checkbox.
    b. The old workflows are discarded. The copy becomes a process type and the workflows must be rebuilt in the Workflow Modeler. See chapter 4.
7. Click *Create*.

You have created a copy of an existing type. To continue editing, proceed as described in chapter 3.3.2.1 from step 8 onwards.

3.3.2.4 Localize Display Name

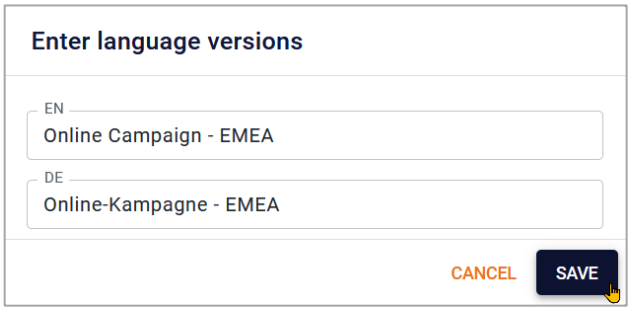
To localize the displayed name of a type, open it by clicking on the ✎-icon in the type list. You create and edit the translations on the *Properties* page. This option is also available directly during the creation process.

1. Locate the *Display name* field in the newly created type.

    There is a 🌐 icon above the text field. The country codes indicate which language is already stored.

2. Click the 🌐 icon to store the job name in multiple languages.

    The *Enter language versions* dialog opens.



3. Fill in the fields for all the languages you need with a translation.
4. Click the *Save* button to apply the changes to the language versions and return to the properties.
5. Optional: Click on an ISO language code next to the globe to switch between the available language versions and set the language of the field as the displayed name.

6. Optional: Use the *Log* button to display the sidebar with the change log and validation result.
7. Click the *Publish changes* button.

   The *Publish changes* confirmation dialog opens.
8. Click *Confirm*.
9. Close the page by clicking on the close box ✕.

You have created the desired language versions. The language versions are now available in the respective interface languages.

3.3.2.5 Delete Type

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Types*.
2. In the table, click on the icon at the end of the row for the type you want to delete.

   The *Delete* dialog box opens.
3. Click *Confirm* to permanently delete the type.

You have deleted the type.

> Note
>
> Deletion of types is only possible if no job or process of this type is currently under way.

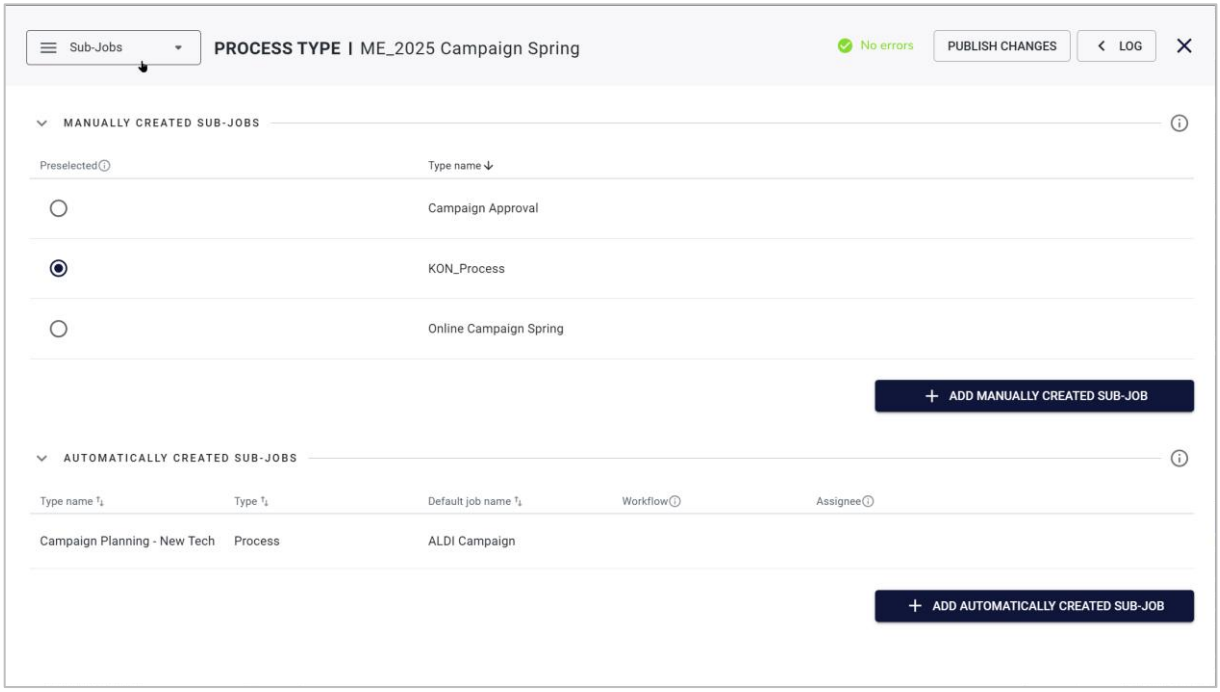
3.3.3 Add a Sub-Job or Sub-Data Object

You can use sub-jobs to structure workflows and make dependencies between different workflows visible. You can use sub-data objects to subdivide a data object or a data object data set. You can specify whether users can add sub-jobs or sub-data objects manually when creating a new job or data object or whether certain sub-jobs or sub-data objects are added automatically.

Prerequisites

- You have defined allowed sub-job types or sub data object types.
- You have already created job types or data object types and linked them to a workflow.

1. Use the ☰ selection menu to switch to the *Sub-Jobs* page while you are creating or editing a job type.



2. In the *Automatically created sub-jobs* area, navigate to *Add* automatically created *sub-job* (or *Automatically created sub-data objects* > *Add sub-data objects*) to specify the sub-types that are automatically added when the job is created.

   This will open a dialog box in which you can make the necessary settings.

When configuring a sub-process, you can specify that the current editor of the parent process is automatically entered as creator of the sub-process.

> Note
>
> The settings required for adding a sub-job, sub-process and sub-data object are identical. The settings are described using the example of a sub-job.

3.3.4 Export and Import of Process Types

Export and import existing process types to simplify and speed configuration of new systems.

Please note that email templates are not transferred during export. You will have to adapt BPMN workflows with a send task after import. Create the email templates again (see chapter 4.4**Fehler! Verweisquelle konnte nicht gefunden werden.**) and assign a recipient or group to the *send task* (see chapter 4.2.5).

3.3.4.1 Export of a Process Type

The export is done as a JSON file. The export includes the following features, rights, and parameters:

- Type properties
- Users, comfort tables as well as variables in connection with the sync to Marketing Planner
- BPMN workflows
- Access permissions
- Input/output parameters for service tasks
- Step names of the simplified view

3.3.4.2 Import of a Process Type

If you have set up and configured a process with BPMN workflow on your test system, it should also be transferred to live operation at some point after thorough testing. The import is done in the creation wizard:

1. Choose > *Administration* > *Overview* > *Datasheet Engine* > *Types*.
2. Click *Create*.
3. Select the *Import from file* option.

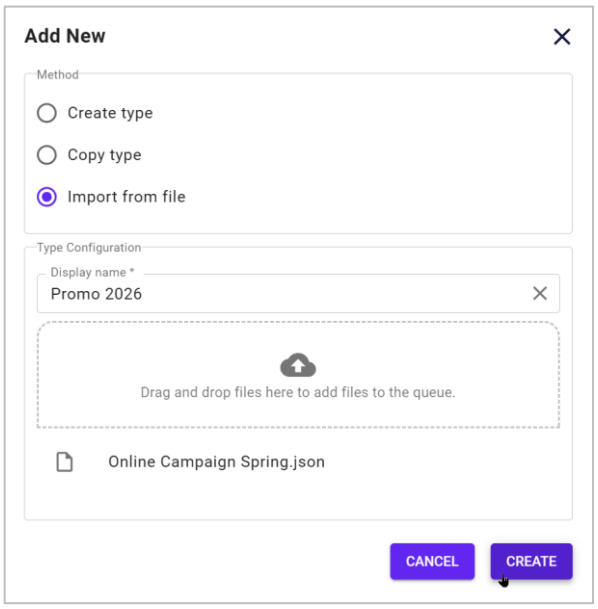
> Note
>
> Imports take place without the data of a user-defined data structure. If the data structure on the import system is incomplete, you will see error messages on the relevant gateways. The process cannot yet be used. The missing data structures must first be recreated and relinked in the imported BPMN workflow.

There are two ways to import a JSON file:

a. Drag the desired JSON file from the file system to the marked upload area. If the file extension is correct and the import process is possible, the file name is listed below the upload area.

b. Another option is to click on the selected area, whereupon the file selection dialog opens with the default file type filter *.json*. In this dialog, you can navigate in the file system and search for the desired JSON file. Select the file for import and click *Open*.

4. Enter a name for the process in the *Display name* field.



5. When you click on the *Create* button, the import is executed and the process is saved under the name you have specified.

   You are on the *Properties* page.

6. To continue editing the process type, proceed as described in chapter 0 from step 8 onwards.

You have imported a process type that was previously exported from another system.

> Note
>
> Imports of processes with synchronization settings only work to a limited extent. This is because the settings for dimensions and Planner layers are ID-based, but these IDs differ from one system to another and cannot be re-mapped. Therefore, a manual re-creation is necessary.

3.4 Assign Workflow

To use a type, you must link the type to a workflow. This ensures that all the steps required for processing a datasheet are carried out.

Note:

- Job and data object types are linked to a classic workflow.
- You create a BPMN workflow for a process.
- You have two options for creating a classic workflow:
- Under > *Administration* > *Overview* > *Data structures & workflows* > *Workflows,* you can create classic workflows that can also be used by other modules. In this case, you must assign one or more workflows to the job or data object types. Please refer to the following section 3.4.1.
- Under > *Administration* > *Overview* > *Datasheet Engine* > *Types, create* a classic workflow that is used exclusively by this job or data object type. Please refer to section 3.4.2.

3.4.1 Assign Existing Workflows

You can assign workflows directly when creating the type or edit the properties of the You have two options: *From scratch* or *with existing workflows*.

- If you select *From scratch*, you have to define a new workflow yourself and have the freedom to design it according to your own needs and requirements. See section 3.4.2.
- If you select *With existing workflows*, you can use an existing workflow. You can select a workflow that has already been created and adapt it if necessary. This option saves time as you do not have to create everything from scratch.

Prerequisite

You have already created a type without workflow. See section 3.3.2.1.

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Types.*
2. In the list, search for the job or data object type that you want to edit and click the pencil icon to the right of the name.

   The type is opened.
3. Switch to the *Workflows* page via the ☰ selection menu.
4. Click on the *Create new* button to create a new workflow.

   For details of how the workflow is created, see the following subchapter 3.4.2.
5. Save the new workflow with *Save*.
6. Use the *Log* button to display the sidebar with the change log and validation result.

   a. Eliminate existing validation errors. The error messages specifically point out errors in the workflow and datasheet so that you can rectify them quickly.
7. Click the *Publish changes* button.

   The *Publish changes* confirmation dialog opens.
8. Click *Confirm*.
9. Close the page by clicking the close box ✕.

You have created a new workflow for the job type or assigned an existing one.

> Note
>
> After you have assigned or customized a classic workflow, you can also close the *Workflows* page immediately (×) without having to publish the changes. In this case, the changes are first collected and published at a later time. How this works is described in detail in chapter 3.9.

3.4.2 Create Classic Workflow

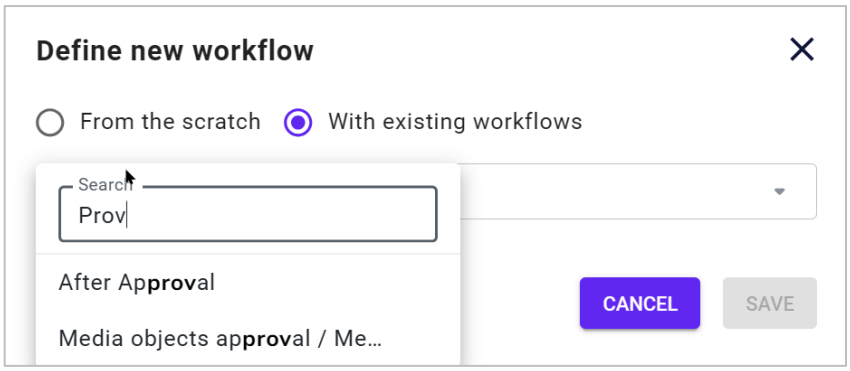
1. Navigate to > Administration > Overview > *Datasheet Engine* > *Types*.
2. Search for the type you want to edit.
3. Move the mouse over the name and click the pencil symbol on the right.

   The Properties page is displayed.
4. Choose Workflows via the ☰-select menu.
5. Click *Create new*.

   The *Define New Workflow* dialog box is displayed.
6. If you want to create a new classic workflow:
   a) Click *From the scratch*.
   b) Enter a name and description for the workflow.

      If you intend to use an existing classic workflow as a starting point:

      a. Click *With existing workflows*.



      b. Select an existing workflow from the drop-down list.
      c. Click *Save*.

You have created the workflow.

3.4.3 Edit classic workflow

1. Click the button to configure the access rights for the steps. See chapter 3.5.1 ff.
2. Click on the plus icon to add another workflow step.
3. Double-click a workflow step or select > ⋮ > *Edit* to edit the settings of the workflow step.
4. Select > ⋮ > *Copy* to copy the workflow step.

5. Select > ⋮ > *Delete* to delete the workflow step.

   The sequence of workflow steps can be adjusted using drag-and-drop:

6. Move a workflow step by clicking and dragging the handle ⁝⁝.
7. Click × to close the Workflows page.

You have created a new classic workflow that is configured to suit this type.

3.4.4 Create a BPMN Workflow for a Process

You can also copy a job type with a classic workflow, remove the existing assignment and assign a BPMN workflow instead. If you select a job type with an existing workflow as described in section 3.3.2.3, the *Assign BPMN workflow* option is displayed. This is a replacement for the legacy *Copy type and assign a BPMN process* function.

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Types*.
2. Select the desired process in the overview list by hovering the mouse over the name.

   Three icons are displayed, representing the functions that can be applied to the type under the mouse cursor.

3. To edit the type, click the pencil icon to the right of the name.

   The *Properties* page is displayed.

4. Switch to the *Workflow Modeler* page via the ☰ selection menu.

The BPMN workflow modeler opens. You can create a new workflow as described in chapter 4.

3.5 Datasheet Layout

You use existing variables and attributes to design datasheets that are used and filled in by users when creating a job, process or data object.

You want to edit the datasheet layout of a specific type. To open the datasheet layout of a type:

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Types.*
2. Search for the type you want to edit.
3. Move the mouse over the name and click the pencil icon on the right.
4. Use the ☰ selection menu to switch to the *Datasheet Layout* page to edit the datasheet of the type.

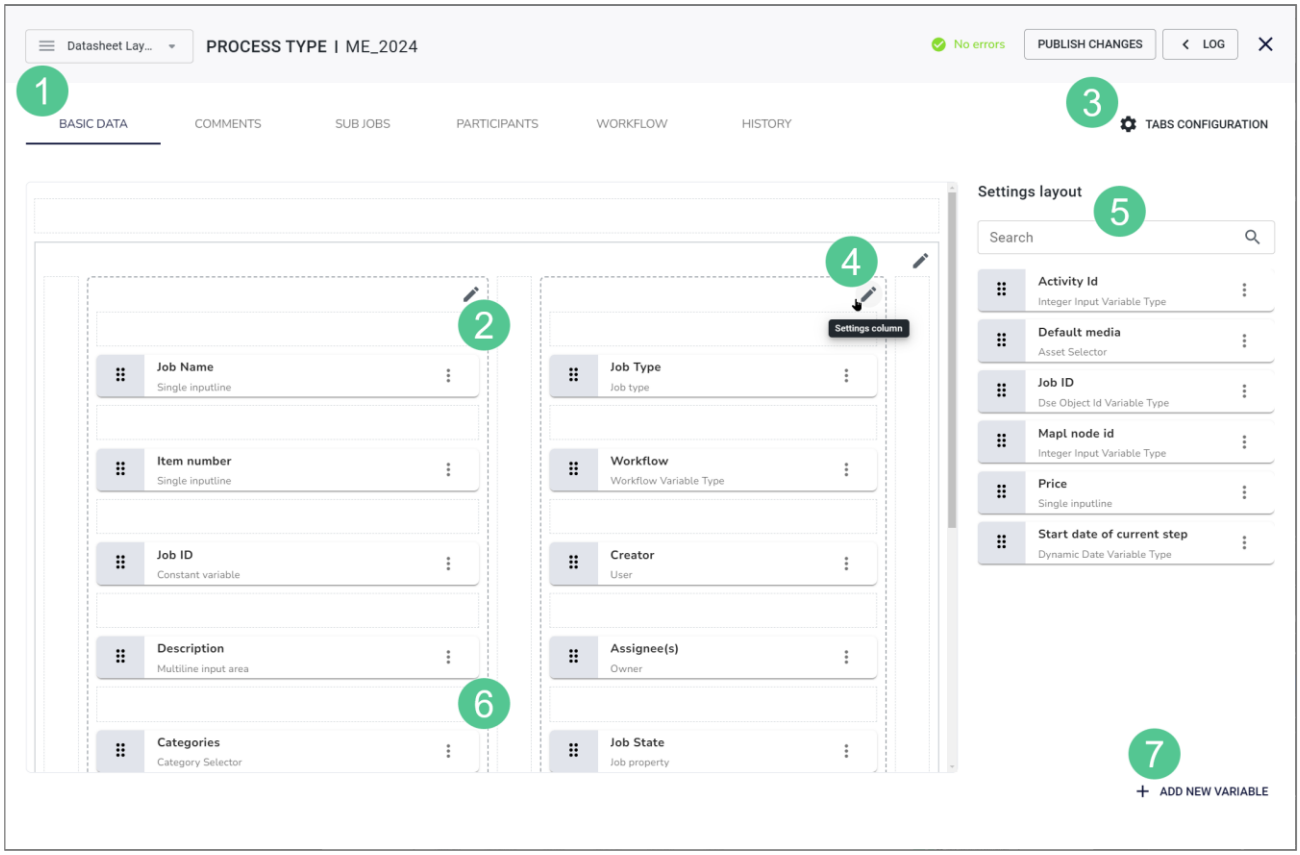
   The datasheet layout is loaded.

You can now customize the layout to your needs and create and place custom variables.

To give users the option of hiding areas that have already been edited later for reasons of clarity, administrators can configure in the settings (3) in the datasheet layout that columns and rows can be minimized by collapsing them.

> Note
>
> The display of the datasheet between jobs and data objects. A job datasheet always includes the tabs *Basic Data*, *Comments*, *Participants*, *History*, and *Workflow*. You can rename, hide and rearrange most of these tabs. Tabs can be hidden using the visibility settings, but you cannot delete them.
>
> Additionally, a job datasheet always contains a job discussion, a Data Hub datasheet can switch between language variants and has the product discussion in the right sidebar instead.

The functional description for the following illustration can be found in the table.^



Number	Description
1	All tabs created in the datasheet are displayed here. Click on a tab name to switch to the corresponding subpage of the datasheet editor.
2	Use the pencil icon at the top right to access the column settings. The column width can be adjusted using the slider. The changes are saved with *Apply*.
3	*Tabs Configuration*: Adjust the order of the tabs after *Basic Data*, add new tabs, rename existing ones or delete them completely. The tab title can also be created in other languages.
4	Use the pencil icon on the outer frame to access the row settings: *Collapsible*, *Separated*. If collapsible, a - appears in front of the row name in the open datasheet to collapse or a + to expand the area.
5	In this pane, you can find the variables that have not yet been placed in the layout. Drag and drop variables from here into the layout onto a vertical placeholder for a new column or onto a horizontal placeholder for a new row above or below the existing layout. You can filter a long list of variables with a search using the search field. Complex layouts with up to four columns with different column widths are possible.
6	This is the area where the datasheet layout is displayed.
7	Click *+ Add New Variable* to create a new variable for use with this datasheet.

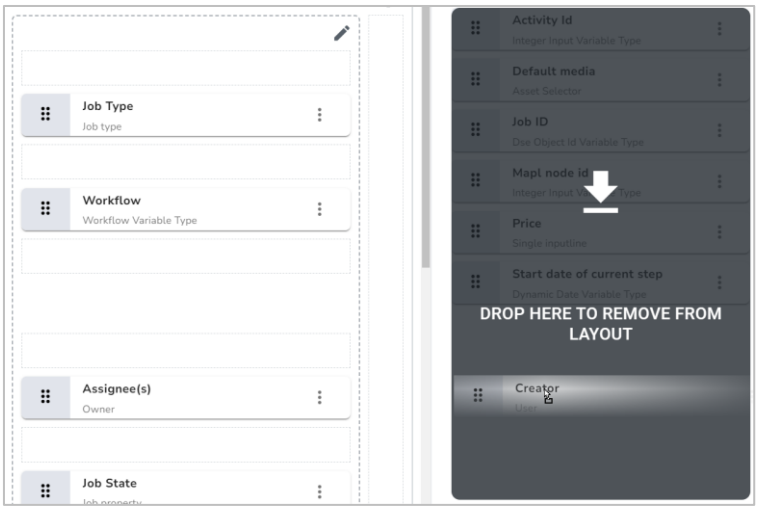
Note

If you want to connect a data object and its data in *the Brand Template Builder* module to the *Smart Group* function, an image must be defined in the *Images* field on the datasheet and the data object must be assigned to at least one theme. Accordingly, the type must provide the system variables *Images* and *Themes*. Take this into account when you create the datasheet.

3.5.1 Edit and Customize Layout

In order to edit and adjust the layout of the datasheet, the following functions are available in addition to the column settings (4):

- Reorder tabs: Click the gear wheel at the top right. This opens a window in which you can rearrange the tabs of the datasheet using drag-and-drop. Click *Save* to apply the changes.
- Sort columns: You can swap the order of the columns by holding the left-hand column by the dashed frame and dragging and dropping it to the new vertical position to the right of the original right-hand column.



- Clear column: To delete a column, drag and drop the dashed frame on the right onto the *Settings layout* area. The dropped frame and its variables are removed from the layout.
- Delete layout: In the same way, drag and drop the gray outer frame back to the right in the *Settings layout* area to clear the layout and redesign it from scratch.
- Context menu: You can access the functions for editing or copying individual variables via the ⋮ context menu on a variable field placed in the layout.
- Arrange variables: You can use the ⁝⁝ handle to reposition an individual variable on the layout via drag-and-drop, or move it back to the *Settings layout* pane to remove it from the layout.

3.5.2 Datasheet Tab Visibility

The access and visibility of tabs and the variables placed on them can be restricted when configuring the type for each workflow step. For this purpose, users are divided into the categories of *Assignee*, *Creator*, *Other participants* and *Anonymous*. Which tabs and variables of the datasheet are visible or editable depends on which category you belong to as a user when opening a datasheet. You can configure the visibility settings for all user categories. These settings apply at the workflow level.

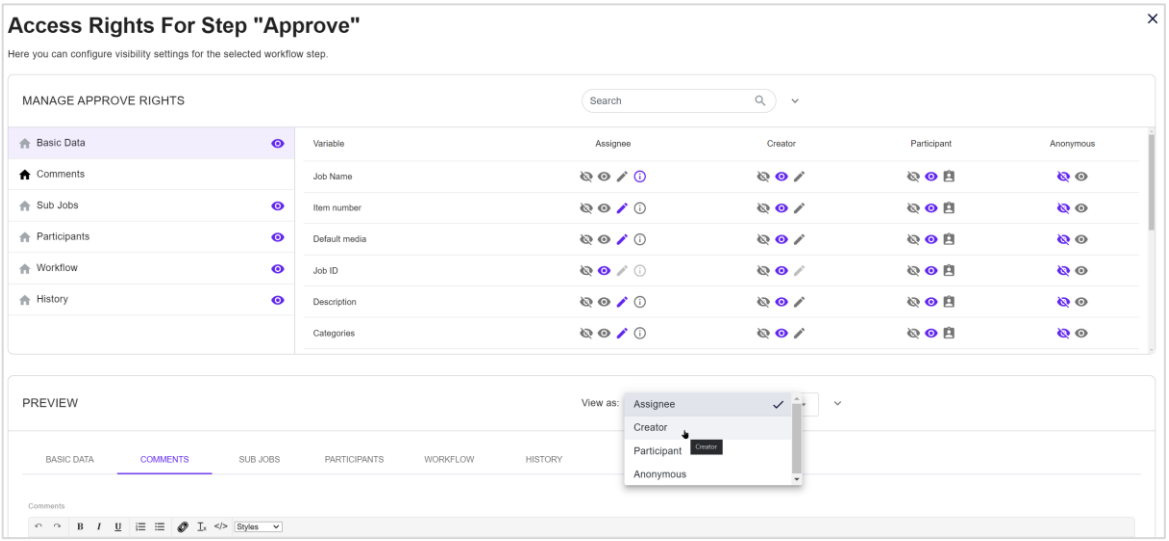
> Note
>
> This section describes how to define the visibility for job and data object types. A description of the visibility of processes can be found in chapter 4.7.

You want to restrict the visibility of the *Sub Jobs* datasheet tab. To edit the visibility of a datasheet tab, go to > *Administration* > *Overview* > *Datasheet Engine* > *Types* and open a job type or make the settings directly when creating a new type. The visibility of a datasheet tab can be defined per workflow step for classic workflows and per user task for BPNM.

For a job datasheet with classic workflows, you can configure the visibility of the individual tabs for each workflow step when creating the type as follows:

1. Switch to the *Workflows* page via the ☰ selection menu.
2. Click the icon in the row of the first workflow step to access the access rights page.

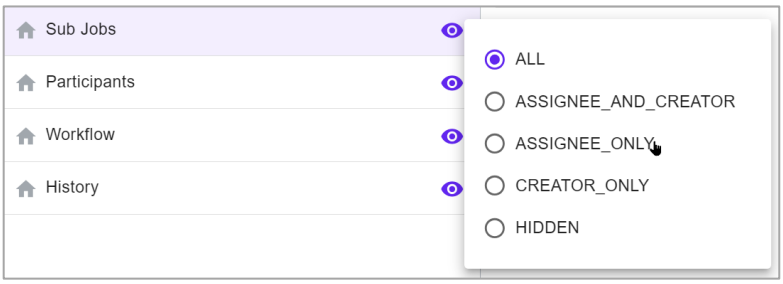
   The *Access rights* page is loaded. Here you can customize the standard tab on the left and view and edit the visibility settings for the variables in the selected workflow step on the right. See Chapters 3.5.4 and 4.7.



The *Access Rights* page shows all tabs and variables, their current visibility and access permissions for the three user categories. With a selection in the *View as* drop-down menu, you can see what the datasheet tabs look like for different participants directly in the preview area below. In the *Creator/Assignee* column, you can select mandatory fields to be filled in.

3. Click on the eye icon of the *Sub Jobs* tab in the column on the left.

   A selection menu opens.



4. Select *ASSIGNEE_ONLY* from the menu.

   This means that you have specified that only *Assignees* can see the tab in the datasheet later.

5. In the *Preview* area, select *View as* one of the three user categories from the dropdown menu.

   Below this, a preview of the tab with placeholder data is displayed. For smaller screens, scrolling down to preview all variable fields may be required.

6. Click the close element ✕.

7. Optional: To adjust visibility for other tabs and in other workflows of the type, perform steps 2-6 again.

You have customized the visibility for a tab of the job datasheet in a specific workflow.

3.5.3 Setting Scenarios for Tab Visibility

Menu selection	Creator sees tab	Assignee sees tab	Participant sees tab
ALL	✓	✓	✓
ASSIGNEE_AND_CREATOR	✓	✓	✗
ASSIGNEE_ONLY	✗	✓	✗
CREATOR_ONLY	✓	✗	✗
HIDDEN	✗	✗	✗

The goal of the new *CREATOR_ONLY* visibility is to make marked tabs visible only to the creator of a job, but not to other users such as assignees or participants. In versions prior to 7.5, all administrators with the *Act like Creator* permission could see these tabs. For each datasheet tab, the visibility is preset to for everyone (*All*).

3.5.4 Set Default Tab

When the user opens a data sheet, the default tab is displayed. By default, the first tab *General is* the standard tab. However, as an administrator, you have the option of changing this setting:

1. Switch to the *Workflows* page via the ☰ selection menu.
2. Click the icon in the row of the first workflow step to call up the page with the access rights.

   The *Access rights* page is loaded. You will see a grey house icon on the left in front of each tab name. If the workflow has already been customized, you will see a black house icon in front of a tab.
3. On the *Access rights* page, in the column with the tabs, click on the gray house icon to the left of the desired new standard tab for this workflow.
4. The house icon turns black and the eye icon disappears, indicating the non-standard status.

You have marked a tab as the default tab for the workflow step. When users open the data sheet, the tab marked as a favorite for the workflow step is opened instead of *General*.

3.6 Managing Variables

You can edit the variables that are assigned to a type or add new variables. After you have opened a type, the assigned variables can be viewed and managed in the *Datasheet Layout* tab.

> Note
>
> If you want to connect a Data object and its data in the *Brand Template Builder* module to the *Smart Group* function, an image must be defined in the *Images* field on the datasheet and the data object must be assigned to at least one category. Accordingly, the type must provide the system variable *Images*. Take this into account when you create the datasheet.

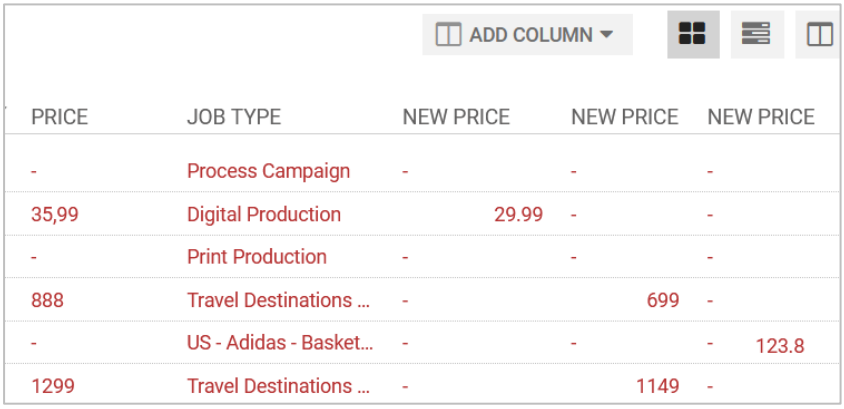
Prerequisite

- You have the permission *Manage Jobs*.

⁝ Menu/Button	Description
+ ADD NEW VARIABLE	You create a new variable for this type.
	You are editing the selected variable.
	You copy the selected variable.
	You delete the selected variable.

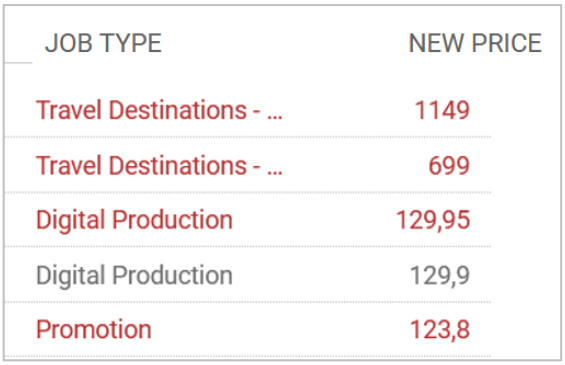
3.6.1 Grouping Variables

Basically, a variable is created for each type individually. The overview in the *Job Manager* module then displays the values of the variables in separate columns. However, this can quickly become confusing if a variable with the same name is used in different job types.



To compare certain values for each job, each process and each data object, you can group variables. It is then possible to display these values in a column in the overview. Examples of this are

- The price of data objects that are derived from different data object types.
- The completion date of jobs that are derived from different job types.



- In order to group variables, the variables in the different types must have an identical technical name and be of the same variable type. In addition, the *Cross-type* option must be activated in the variable settings.
- If you want to display grouped type spanning variables in a column, you will find them in the job overview in the *Add column* menu under *General*. They are not listed individually under their associated job names..

A variable column can be deactivated again by hovering over any column title on the job overview page until the ▾ menu becomes visible on the side. There, under *Columns*, remove the checkbox in front of the column title.

3.6.2 Visibility of Variables

> Note
>
> This section describes how to set visibility for job and data object types. For a description of the visibility of processes, see Chapter 4.7.

As an administrator, you can define the access rights for the variables. This allows you to determine which variables are visible and editable for which users.

Who is this Functionality Intended for?

You work in an organization that has strict compliance requirements and you want only authorized people to have access to certain variables.
By customizing the variables, you can maintain data sovereignty and limit the flow of information to what is necessary for each role. If you disable the visibility of a variable, it will not be displayed to the corresponding user category. If it is only visible, it can be viewed but not changed. Separate control per variable and user category allows for a fine-grained access control matrix. This allows you to adapt workflows and datasheets very specifically to the required access rights. Overall, the definition of visibility and permissions thus serves the secure and role-based control of the information flow within the individual variables.

You must make these settings separately for each workflow step.

Three user categories are distinguished:

- *Creator/Assignee*: The creator is the user who created the job or data object type. The assignee is the user responsible for the workflow step.
- *Participants*: Other participants are users who have been invited to the job or data object type.
- *Anonymous*: All other users who do not belong to one of the user groups mentioned are referred to as anonymous.

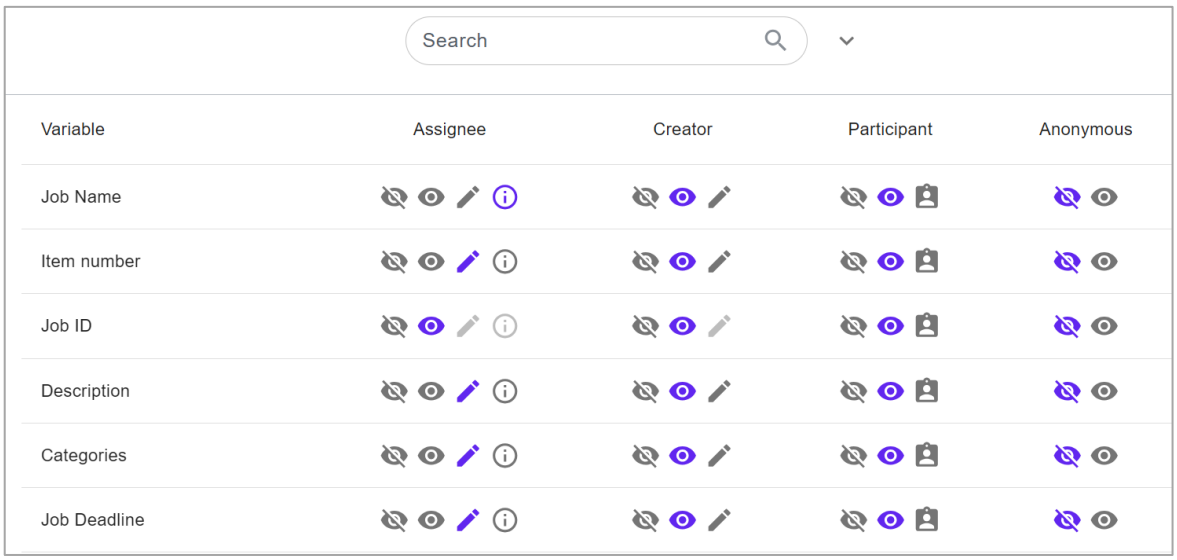
You can set the visibility for each of the three user categories individually. Switch variables individually to visible or hidden. You can also set editability or write protection so that the variable field is read-only but cannot be changed by the user. For *creators/owners* , you can mark mandatory fields as the fourth option, which must be filled in before the workflow step can be saved and finished.

The visibility in the datasheet can be configured individually for each workflow step. A view opens with an overview of all variables used on the individual tabs. On the right in the *Access Rights* window, you will find the list of all variables of the selected tab. If this list is too long for you, there is a search box above it to narrow down the results.

In the following section, you will learn how to set the visibility of variables individually for each workflow step in the datasheet:

1. Navigate to > Administration > Overview > *Datasheet Engine* > *Types*.
2. Open the desired job type by clicking on the pencil icon at the end of the row of the type.
3. Switch to the *Workflows* page via the ☰ selection menu.
4. In the line for the workflow you want to customize, click the icon to edit the access rights and visibility of variables.

   You have opened the *Permissions* page for the selected workflow. After opening, you will see the variables of the *General* tab by default.



5. If you click on a tab in the column on the left, the variables placed on this tab are displayed in the table on the right.
6. To change a variable, click on the corresponding icon. The icon changes its color.

You can customize the access rights to the variable fields for each datasheet tab under the four columns *Assignee*, *Creator, Participant* , and *Anonymous* .

You can set the visibility of the variables and their editing options by clicking on the following icons :

Name/Symbol	Description
	The variable is displayed and can be edited.
	The variable is displayed but is read-only and may not be edited.
	The variable is set as a mandatory field and must be edited. You can see this in the type preview area and the later editor on * after the variable name in the data sheet.
	The variable is hidden.
	Activates all rights as for the owner.

You can individually set the visibility of the variables and other attributes for the three user categories using the following table. Inactive settings are shown as gray icons. The options, presets and previous adjustments you have chosen are highlighted in color.

User type	Visibility	Editable	Mandatory field
Creator/Owner *(Creator/Assignee)*	Adjustable: Visible (default) or hidden	Adjustable: Editable, read-only	Adjustable: Mandatory field or no mandatory field (default)
Participant	Adjustable: Visible (default), hidden or as for owners	Adjustable: Editable, read-only or same as for owners	Not adjustable
Other participants *(Anonymous)*	Adjustable: Visible or hidden (default)	Not adjustable	Not adjustable

Finally, you must run *Publish changes* for the customized type.

3.6.3 Editability of Variables

In addition to visibility, you can also set the editing options for each variable and user category:

- Editable: The variable can be edited and saved.

- Read-only: The variable is visible but read-only. It cannot be changed.
- Mandatory field: The field must be filled in. The field is marked with an asterisk * in the user interface.

With the help of the previous table, you can individually determine for each field and each user category whether it should only be displayed, editable or mandatory.

3.7 Workflow Messages

Messages are generated automatically and sent for actions such as forwarding to the next workflow step or inviting a user to a job, for example.

Prerequisite

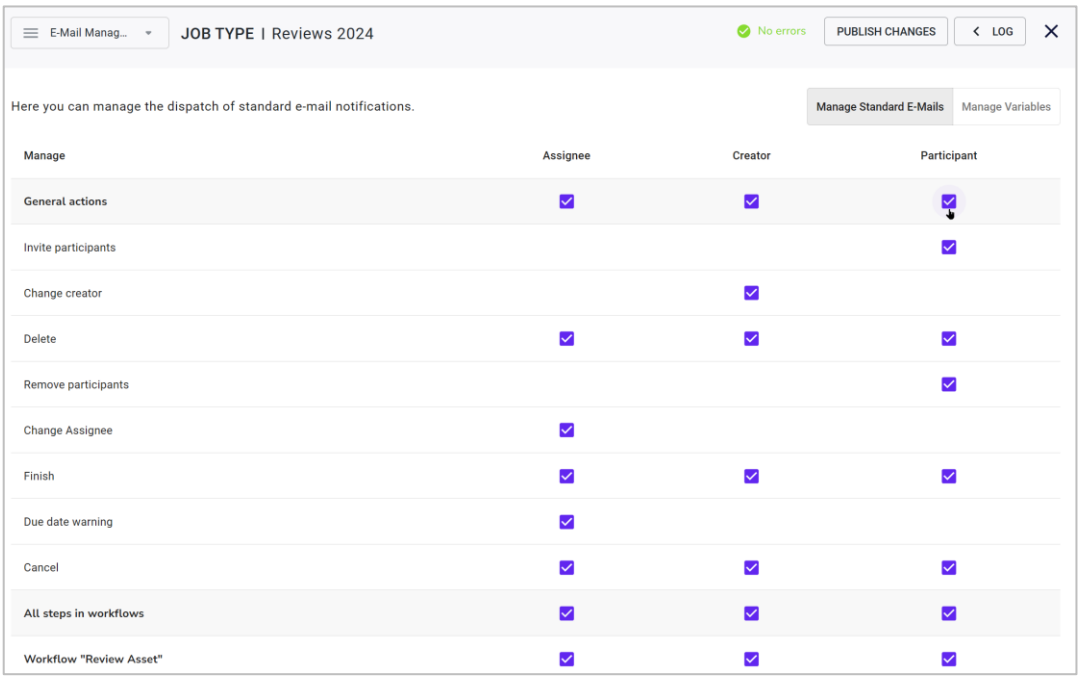
- You have the permission *Manage Email Notification*.

3.7.1 Manage Standard E-Mails

1. Navigate to > Administration > Overview > *Datasheet Engine* > *Types*.
2. In the table, click the ✎ icon at the end of the line for the type whose properties you want to edit.

   The Job type *properties* page is displayed:
3. Use the ☰ selection menu to switch to the *E-Mail management* > *Manage Standard E-Mails* page.

You have opened the *Manage Standard E-Mails* page.



3.7.1.1 Who Should be Notified by E-Mail?

On the *Manage Standard E-Mails* page, you can define which people should be notified of events by e-mail for the open type. For classic workflows, the *All steps in workflow* area is also available for selection in addition to *General actions*. The users are divided into three different user categories.

User Categories

- *Assignee*: The user who is responsible for the workflow step.
- *Client*: The user who created the type.
- *Participants*: Users who have been invited.

Events that Trigger Sending an E-Mail

- General actions: Complete, Invite participant, Remove participant, Change assigned editor, Cancel, Due date warning, Delete, Change creator
- Workflow-specific actions: Forward to the next step, return to a previous step

Activate or deactivate the corresponding checkboxes to configure the sending of messages. Click the checkbox in the gray title bar if you want to completely deactivate the notifications for this action and this user category. Please note that the user who triggers an event will not receive a message.

3.7.2 Manage Variables

The available variables are listed under this tab. You can define which variables appear in which order in the standard messages under the *Job variables* section.

To rearrange and sort the variables: For activated variables (checkbox checked), you can change the order individually using the = grab tool.

The list contains the following columns:

Name	Description
Status	Activate the checkbox to adopt the variable or its value in the system messages.
Variable	This displays the name of the variable.
Tab	This displays the tab on which the variable is placed.

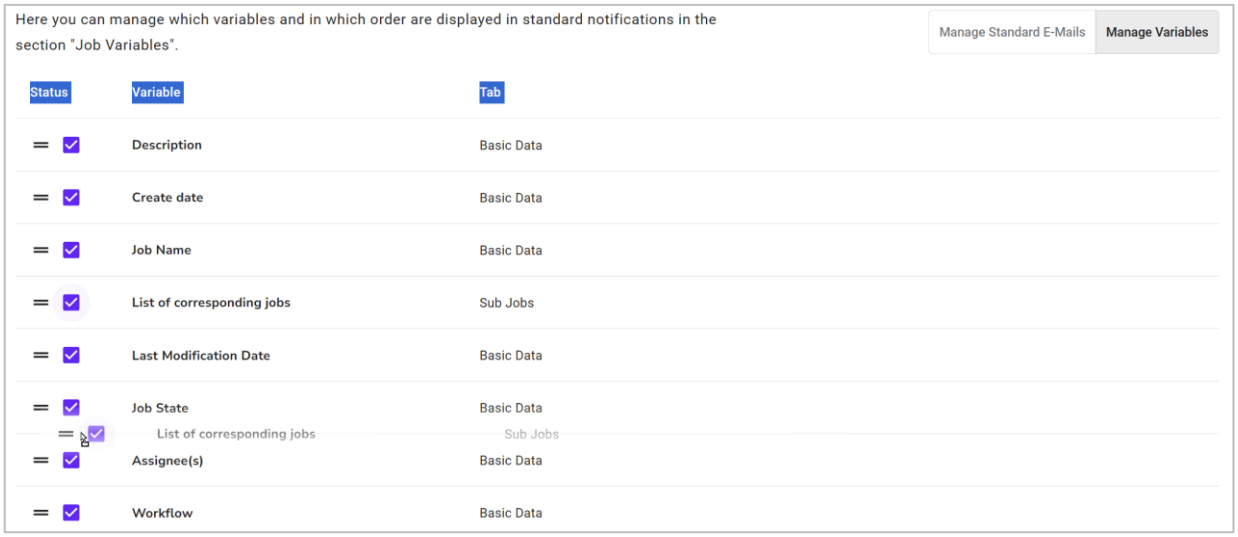
1. Navigate to > *Administration* > *Overview* > Datasheet Engine > Types.
2. In the table, click the ✎ icon at the end of the row for the type whose properties you want to edit.

   The *Properties* page of the job type is displayed:

3. Use the ☰ selection menu to switch to the *Manage Standard E-Mails* > *Manage Standard E-Mails* page.
4. Go to the *Manage Variables* tab.

   Click the box in front of a variable to display it in the standard notifications.

   This activates the variable for standard messages.



5. To change the order, drag an element with the = grab tool as shown in the figure and drop it in the desired new position.

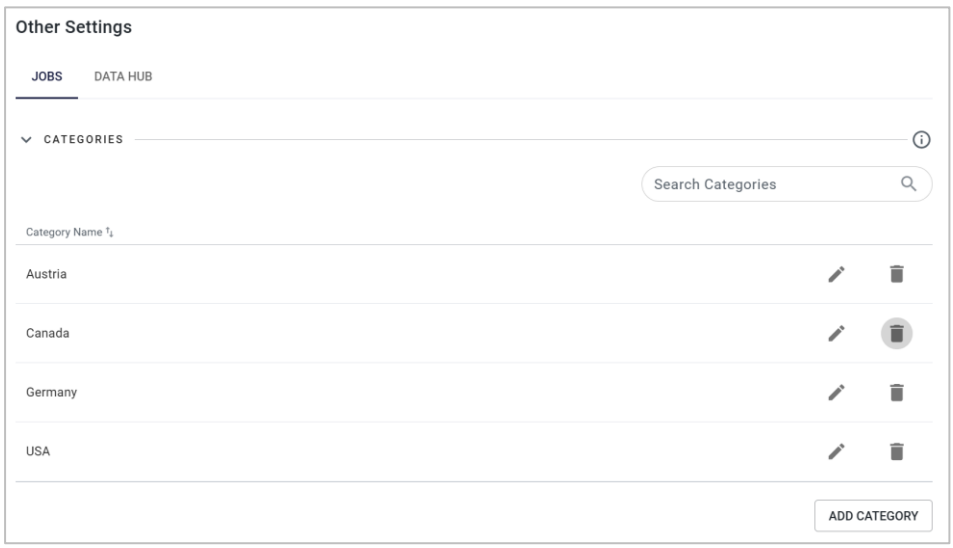
You have defined which variables appear in standard messages under the job variables.

3.8 Other Settings

In the section > *Administration* > *Overview* > *Datasheet Engine* > *Other Settings* and its three subsections *Categories*, *General Settings* and *Other Settings* different settings for types and data objects can be configured globally.

3.8.1 Categories

For a better overview, you can create categories and assign types to one or more categories. When users later create a new job, process, or data object, they select the desired type in an overview organized by category. You can access the categories settings via
> *Administration* > *Overview* > *Datasheet Engine* > *Other Settings*. In the *Jobs* or *Data Hub* tab, expand the *Categories* section with the right arrow › to see all categories, browse and filter them further.



3.8.2 Creating a Type Category

Use case: You want to create the category *Photo shootings* for the Job Manager module.

Prerequisite

- You have the permission *Manage Jobs*.

Step by Step

1. Click Navigate to > Administration > Overview > *Datasheet Engine* > *Other Settings*.

   You are on the *Jobs* tab.

2. Open *Categories* by clicking the right arrow ›.
3. Select the *Add Category* button at the bottom of the list.

   This will open an *Add Category* dialog box.

4. Enter *Photo shootings* in the Name input field.
5. Optional: enter the category name in other languages.
   - To accomplish this, click on the globe icon. This opens the *Enter language versions* dialog box.
     a. Enter the languages that have not yet been translated.
     b. Finish the input with Save.
6. Exit the dialog box with *Save*.

You have created the *Photo shootings* category for the Job Manager module. You can assign existing and newly created job types to this category.

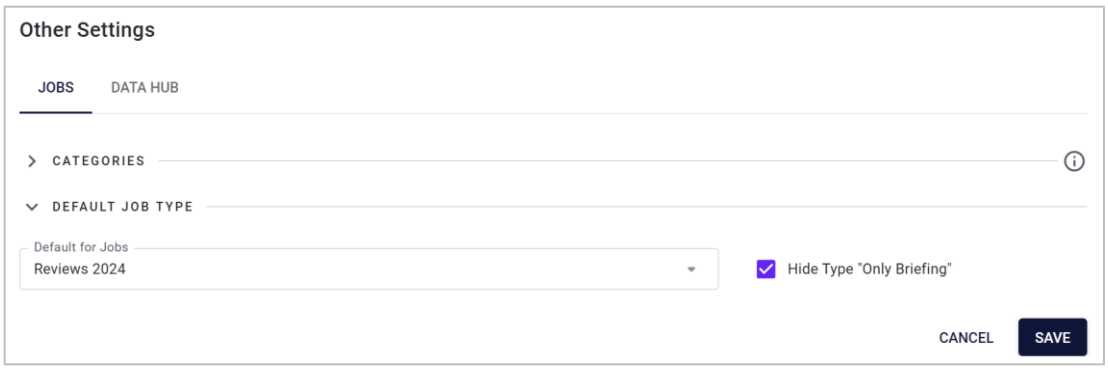
3.8.2.1 *Default Type*

Create the default type for the *Jobs* and *Data Hub* modules. The type defined as the default is then displayed under the category *Standard* when you create a new job, process, or data object. You can also define whether the *Only Briefing* type can be used in the individual modules. With the help of the *Only Briefing* type, requirements for a job, process, or data object can be described simply, without having to configure an extensive datasheet or select a workflow.

Prerequisite

- You have the permission *Manage Jobs*.

1. Navigate to > Administration > Overview > Datasheet Engine > Other Settings.
2. Then select the tab for *Jobs*.



3. Click the right arrow › to expand the *Default Job Type* section.
4. Make a selection from the dropdown menu *Default for Jobs.*
5. Optional: Check the box for *Hide Type "Only Briefing".*
6. Click *Save*.
7. For the changes to take effect, you have to publish them under > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance*.

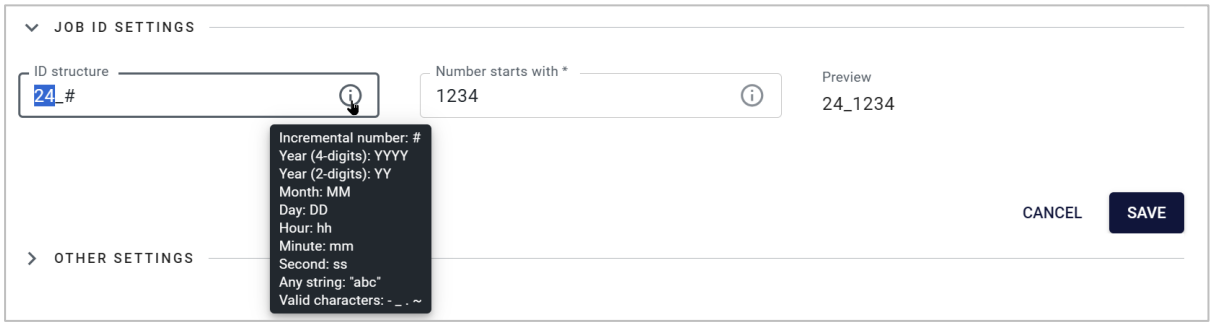
3.8.2.2 *Editing the Format of the Object ID*

Select the *ID structure* text field to edit the format of the object ID for a module.

Name	Description
Possible inputs for ID structure	Enter an ID configuration in the box. The provided tool tip (i) provides you with formats and examples. It lists the characters that you can use for the object ID format, such as "YY" for two-digit year numbers or “DD” for days, for example, and special characters permitted.
Number starts with	The next number used (the ID used last + +1) is entered in the input field automatically. The newly defined object ID format is applied from the entered number onwards. You can enter another higher number. Your entry is checked. If you enter an invalid number, you are informed about this by an info message. In the provided tooltip (i) you can find additional information.
Preview	This displays a preview of the current format of the object ID.

3.8.2.3 *Configuring the Object ID*

When you create a datasheet, a unique ID for the job, processes, or data object is assigned automatically. Note that processes are managed like jobs in the *Job Manager* module. Choose > *Administration* > Overview >Sub *Datasheet Engine* > *Other Settings* > *Jobs* > *General Settings* to determine the format based on which the object ID for jobs and processes is created. In the *Data Hub* tab, you will find identical settings for data objects.



Prerequisite

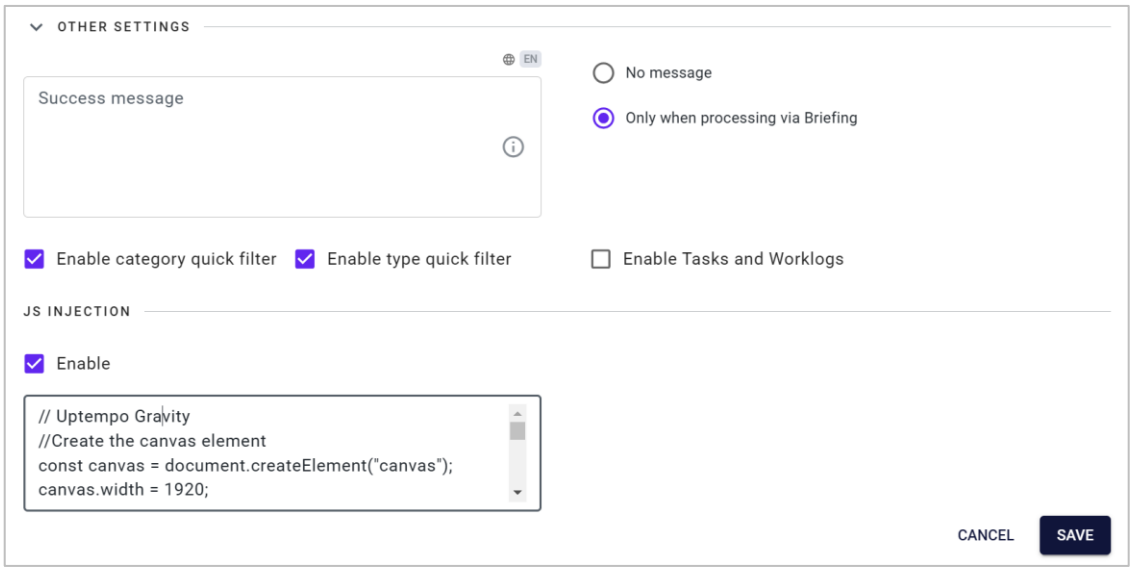
- You have the permission *Manage Jobs*.

Note: A newly defined object ID format is applied as of the next incremental number (#) that is used. Unique IDs that have already been assigned for datasheets remain unchanged. The newly defined formats (a YYYY-MM-DD time stamp, for example) are attached to these IDs.

3.8.3 Sub Category: Other Settings

For the *Only Briefing* type, you can define the message text when a job or data object is forwarded.

1. Click > *Administration* > *Overview* > *Datasheet Engine* > *Other Settings*.
2. Select the tab *Jobs* and click the right arrow to expand the *Other Settings* section.



On this page, you can access the setting options described starting with section 3.8.3.1 below.

Prerequisite

- Use of the *Only Briefing* type is allowed.

3.8.3.1 Success Message

From the dropdown list or input field, you can define which message (if any) is sent when jobs or data objects of the type *Only Briefing* are forwarded:

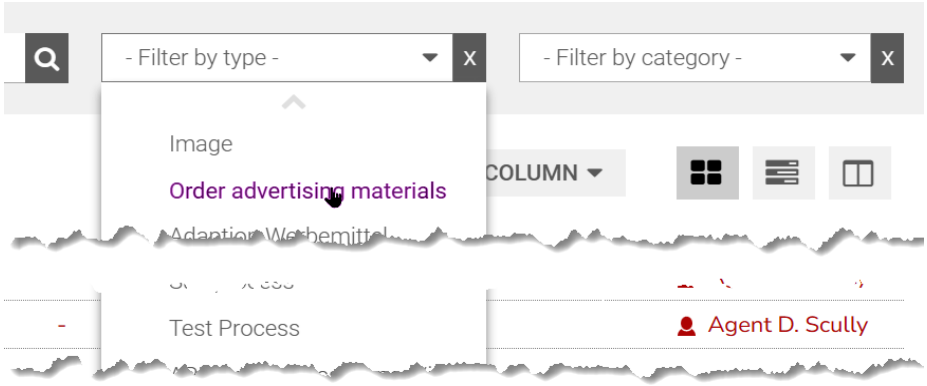
- *No message*: No success message is sent when the item is forwarded.
- *Only when processing via Briefing*: The message stored in the Success message text field is sent when the item is forwarded.

3.8.3.2 Filters

You have the option of activating filters, which are then available in the job overview.

1. If you set the checkbox at *Enable category quick filter,* user can filter by type categories in the job overview. Note: The filter only shows jobs that have the Categories system variable placed and filled on the datasheet.
2. Select the checkbox at *Enable type quick filter* to allow type filters in the overview of the respective module for all users.

The filter options *Filter by type* and *Filter by category* for jobs are enabled.



3.8.3.3 Tasks and Worklogs

If you want to access the navigation for *tasks* and *time management*, activate the checkbox at *Enable Tasks and Worklogs*.

> Note
>
> In version 7.5, it is recommended not to activate the checkbox for *Enable tasks and Worklogs*, as both functions are not yet fully available. If you have any questions, please get in touch with your Uptempo contact person.

All changes on this subpage are applied by clicking *Save*.

3.8.3.4 JS Injection

To cover functions that are not supported by the standard product, it is possible to include code. Follow these steps to enable global integration of JavaScript in the *Jobs* module:

3. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Other settings.*
4. Click the right arrow › to expand the *Other settings* subcategory.
5. Activate the *Enable* checkbox in the *JS Injection* field.

6. Copy your specific JavaScript code into the text field.
7. Click *Save*.

The JavaScript integrated in this way is executed every time users call up the page with the job overview Separately, each job type can have its own JavaScript stored in the *JS Injection* field in the datasheet. The JavaScript in the job type is always executed when a job based on it is opened.

> Note
>
> No syntax check is available for the *JS Injection* field. Do not write code directly in the text field, but use an external editor. It is advisable to test the code in a runtime environment before executing it.

All changed settings in this chapter 3.8.3 must be published on the > *Administration* > > Overview > *Datasheet Engine Maintenance* > *Publish changes* page to make them visible in the module *Jobs*.

3.9 Publishing changes

Changes to a type, process, or data object can be displayed and published individually by clicking on the bell icon with the *Changes* button. As long as no publication of changes is pending, the button remains grayed out. After publishing, the floating sidebar disappears again automatically or can be hidden manually at any time by clicking ×..

All the changes made in the type configuration are grouped together and can be carried out simultaneously at a time that you can specify yourself. This does not only mean the changes to the types themselves, but any changes and adjustments within the datasheet engine. This includes, for example, datasheet layout, categories, task templates or the variables.

Associated tasks

- Publishing changes immediately, page 81: You can publish changes immediately at all times. You can do so even if you have already scheduled a publication.
- Scheduling a publication, page 82: If you want to publish the changes at a specific time, you can enter this time. Administration then carries out the publication at this time. You can schedule a publication even if there are still no changes in the type configuration.
- Changing a scheduled publication, page 83: If you want to perform the publication at a different time, you can edit the time.
- Canceling a scheduled publication, page 83: If you no longer want to perform a publication, you can cancel it.
- Discarding changes, page 84: If you would rather not publish the changes that have been made, you can delete them. Note that you can only undo all the pending changes, not just one.

Prerequisite

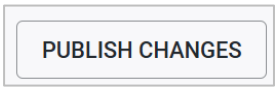
- You have the permission *Manage Jobs*.

Attention!

You have to fix all formal errors before you can publish a newly created or modified BPMN workflow. Publishing with formal errors could permanently damage the job type.

3.9.1 Publish Changes Directly

You publish the changes for each type separately.



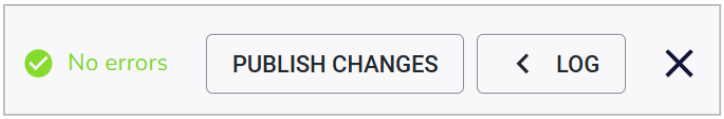
All changes in connection with this type will be published. Please note that you cannot schedule such a publication.

> Note
>
> Note that publishing changes may lead to impact performance for other users. Therefore, we recommend publishing the collected changes for a larger number of types at a time when system utilization is low. Schedule this at a time when there are no or only a few users working in the system.

3.9.2 Validation

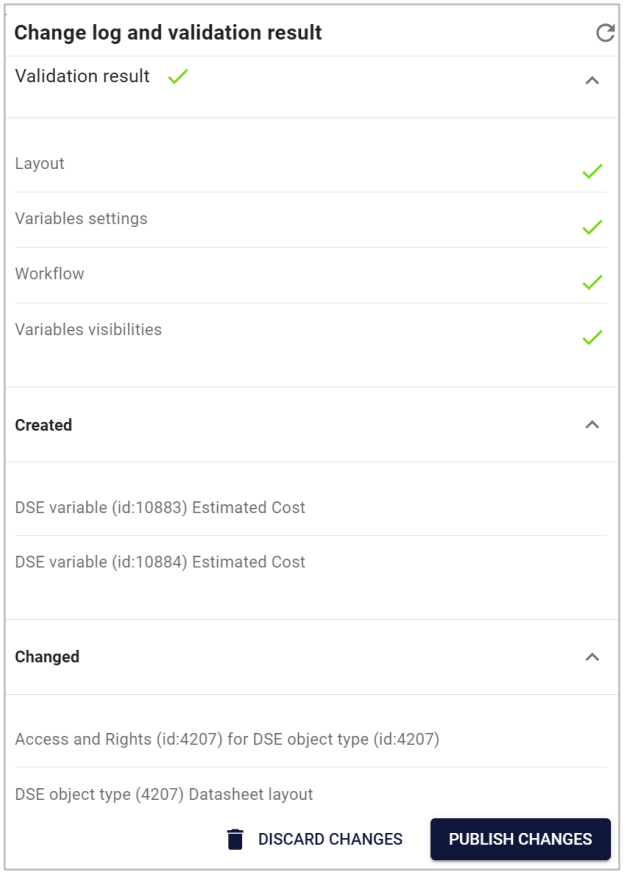
The types are validated automatically within the respective creation dialog. When you create a new type or modify an existing type, you can use the *Log* button to display the changes and validation information on the side panel under the the collapsible and expandable *Validation result* area. Errors that need to be addressed are briefly described and highlighted in red.

As soon as changes have been made to the type, the *Publish changes* and *Log buttons* are displayed in the top right-hand corner. In the event of a validation error, the *Publish changes* button is initially missing.



1. Click the *Log* button to find out more about the exact changes and possible validation errors.

   The side panel with the changes is displayed.



2. After corrections and adjustments, click the ↻ icon in the *Change log* panel to validate the type again.

   You have validated the type and can publish it.

3. Optional: Click the *Publish changes* button to publish the type immediately.
4. Optional: Click the *Discard changes* button to return to the last saved status or to undo all changes made so far.

   You will be prompted for final confirmation.

Only after all validation errors are fixed, can you publish the type using the *Publish* button.

3.9.3 Publishing All Changes Immediately

You can publish changes immediately at all times. You can do so even if you have already scheduled a publication.

> Note
>
> For BPMN workflows, changes to be published are only created when you have changed the way the workflow works. For example, by adding service tasks or user tasks. Nonrelevant changes include changing labels of decisions and step names of simplified workflow steps or moving elements in the editor without changing their order or number.

Prerequisite

- Changes have been made that have not yet been published.

Publish Changes of a Type Immediately

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Types*.
2. Click the pencil icon for the type whose changes you want to publish.

   The editor opens.
3. Click the *Publish changes* button.
4. Click Confirm in the *Publish changes* dialog.

The changes are published. After clicking the *Publish* button, both server-side confirmations and error messages are briefly displayed as browser notifications.

Publishing Accumulated Changes Immediately

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance* > *Publish changes*.

   The list of pending changes is displayed.
2. Click *Publish*.
3. Click *Confirm* in the *Publish changes* dialog.

All pending changes are published.

Note

You will be notified of existing errors in the type configuration on the overview page under > *Administration* > Overview > *Datasheet Engine* > *Maintenance* > *Publish changes*. From there, you can open each type with warnings and errors directly via a deep link to resolve the problem. Only when all pending changes are marked with a green checkmark, can you publish them immediately or at a later date.

3.10 Schedule Publishing

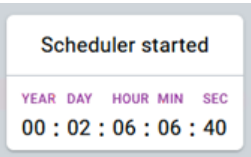
If you want to publish the changes at a specific time, you can enter this time. The back-end system then performs this task on schedule. You can schedule a publication even if there are still no changes in the type configuration.

Note

Publishing changes may lead to impact performance for other users. We therefore recommend publishing changes at a time when the system load is lower.

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance* > *Publish changes*.

   The list of changes is displayed.
2. Click in the date field.
3. Enter the date and time of the publication and confirm it with the Apply button.
4. Click *Schedule Publishing*.
5. In the confirmation dialog, *click Confirm*.
6. A countdown is displayed.



You have scheduled the publication. The changes will be published at the scheduled date and time.

Note

The countdown can only be started after all listed validation errors have been eliminated.

3.10.1 Changing a Scheduled Publication Time

If you want to reschedule a scheduled publication that has already started, you can edit the time.

Prerequisite

- You have already scheduled a publication.

Changing a Scheduled Publication

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance* > *Publish changes*.

   The list of pending changes and the publishing time are displayed.
2. Click *Cancel Scheduling* and then *Confirm*.
3. Use the date picker to enter the new time of publication and confirm it with the *Apply* button.
4. Click the button *Schedule Publishing*.
5. Click *Confirm* to close the dialog.

You have edited the publishing time. The changes are published at the adjusted date and time.

3.10.2 Canceling a Scheduled Publication

If you no longer want to perform a scheduled publication, you can cancel it.

Prerequisite

- You have already scheduled a publication.

Canceling

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance* > *Publish changes*.

   The list of changes and the publishing countdown are displayed.
2. If you have changed your mind, select *Cancel Publishing*.
3. In the confirmation dialog, *press Confirm*.

The publication is now canceled. The changes therefore do not take effect.

3.10.3 Discarding Changes

If the changes you have made are not to be published, you can delete them completely. Please note that you cannot undo individual changes at this point, only all pending changes. To discard individual type changes, you must open the type, click on *Log* and then *Discard changes*.

> Warning! Data loss!
>
> If you discard changes, all the unpublished changed in the type configuration are discarded. You cannot undo this step!

Prerequisite

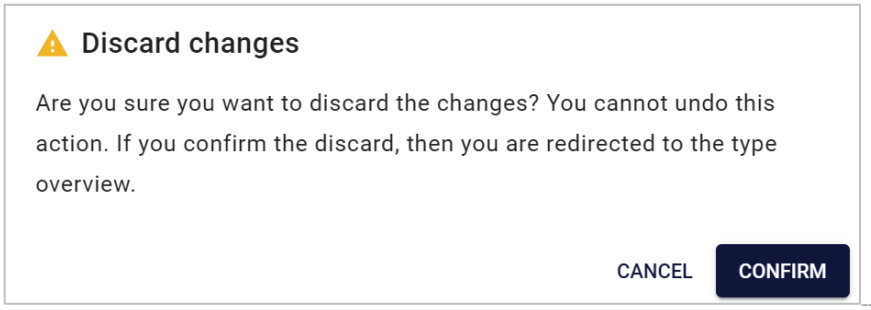
- An administrator has added changes that have not been published yet.

Discard Changes of a Single Type

1. Click > *Administration* > *Overview* > *Datasheet Engine* > *Types*.
2. Select the pencil icon for the type whose changes you want to discard.

   The editor opens.
3. Open the change history in the side panel by clicking on *Log*.
4. Click the *Discard changes* button.

   The following confirmation dialog opens:



5. Click *Confirm*.

   Once the process has been successfully completed, a pop-up appears.
6. Click anywhere outside the window on the screen to close the pop-up window.

You have dicarded your changes in the type configuration.

Discarding All Changes

1. Navigate to > *Administration* > *Overview* > *Datasheet Engine* > *Maintenance* > *Publish changes*.

   The list of pending changes is displayed.

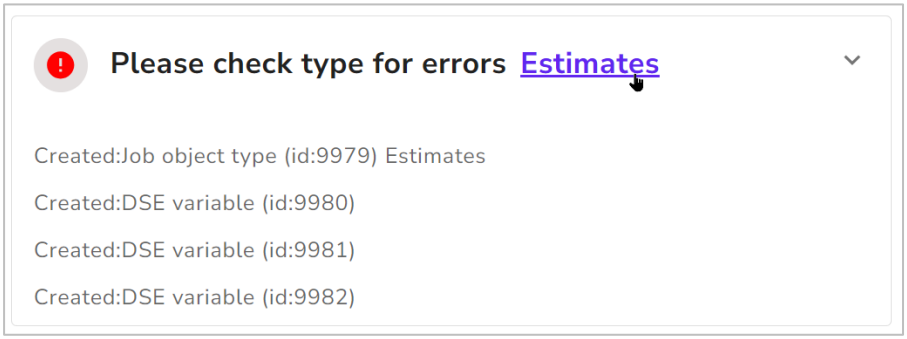
2. Click *Discard Changes*.
3. In the confirmation dialog, *press Confirm*.

All unpublished changes in the type configuration are deleted. The page displays “No changes”.

3.10.4 When Errors Prevent Publication

When setting up a new system, thousands of database changes quickly accumulate. If several people are responsible for maintaining the types in the system, in such a case you will of course not immediately recognize which and above all, whose changes could not be published because they are incorrect.

The display of remaining validation errors on the *Datasheet Engine* > Overview > *Maintenance* > *Publish changes* page supports you in troubleshooting. However, don’t let this error list worry you, because it doesn’t mean that all type customizations are lost. From here you can open any type with warnings and errors directly via a deep link and edit them to resolve the problem.



Continue as described above in section 3.9.2.

After a scheduled publication, you can be notified of the publication status on the *Publish changes* page. If the page is empty after execution, and you see the *No changes* notice, everything worked.

4 BPMN Workflow

This chapter explains how to create BPMN workflows for processes. You can create a BPMN workflow in the BPMN Editor.

Don't be afraid of the initial complexity. Take your time to learn the ropes. If you had only a few building blocks at your disposal before, entirely new possibilities open up now. With BPMN, you have numerous design levels and elements at your disposal. The following chapters can help you with the conversion from classic workflows to BPMN. A good way to do this is to use the *Copy type* option when creating a new type, select a classic job, but enable the *Assign BPMN Workflow* option. See Chapter 3.3.2.3.

Open the BPMN Editor

1. Click > *Administration* > *Overview* > *Datasheet engine* > *Types*.
2. Click the pencil icon for editing a process type.
3. Select *Workflow Modeler* from the ☰ dropdown menu.

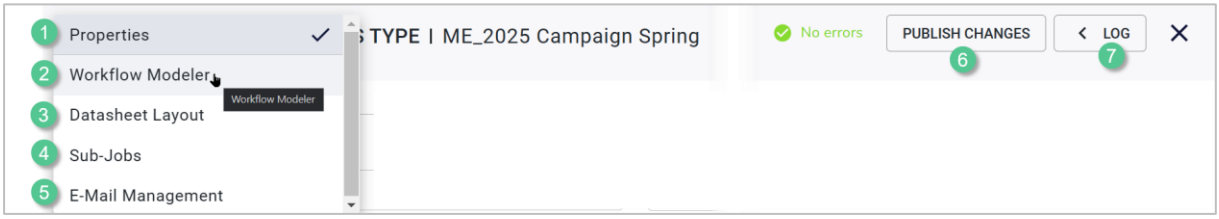
The BPMN editor opens. The following chapter describes the layout of the editor.

4.1 Structure of the Editor

The BPMN workflow modeler allows you to create detailed workflows. Begin with the *Start Event* and add tasks step by step to map the entire process. The datasheet editor consists of the following sub-pages for a BPMN process type:

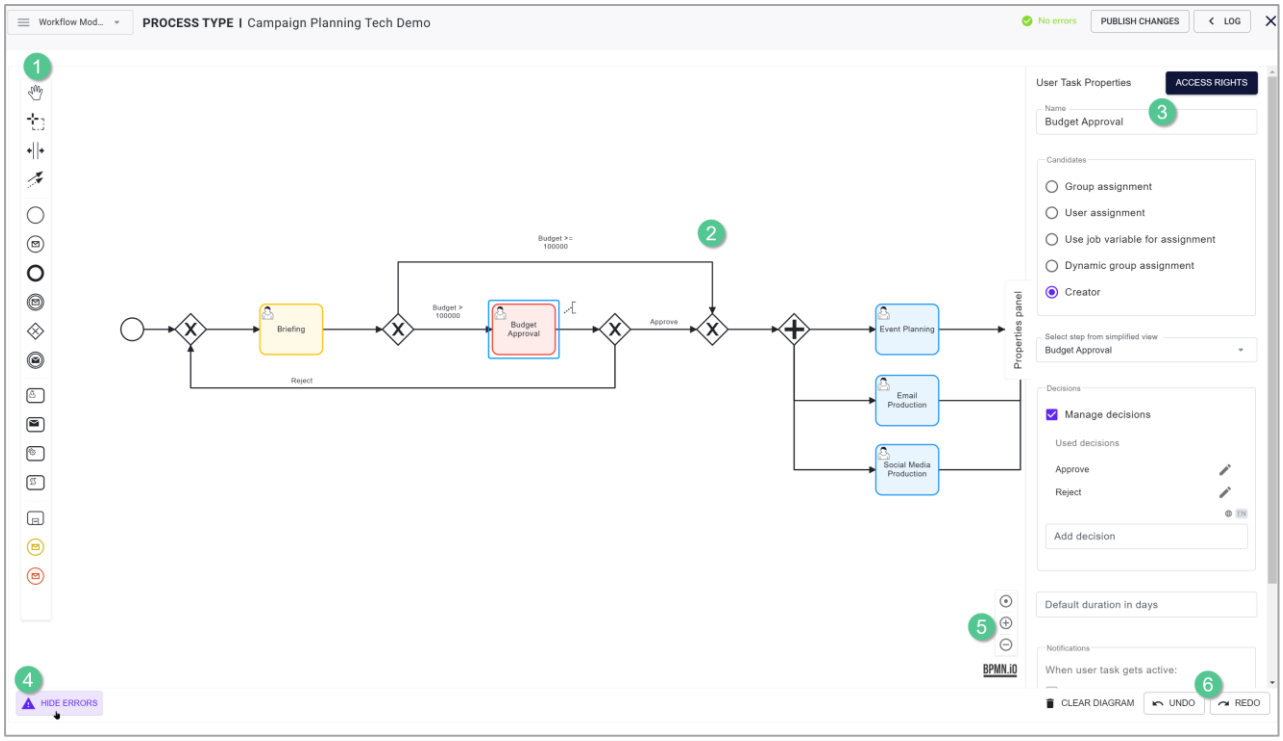
- Properties: All properties of the type are entered or edited here (1). The Properties page is described in Chapter 3.3.1.
- Workflow Modeler (Chapter 4.1.1, 2)
- Datasheet Layout (Chapter 3.5, 3)
- Sub-jobs (chapter 3.3.3**Fehler! Verweisquelle konnte nicht gefunden werden.**, 4)
- E-Mail Management: Editor for creating e-mail templates for sending tasks (Chapter 4.4, 5)

Changes button: As soon as changes have been made, the previously grayed out *Changes* button becomes active (5).



As soon as changes have been made, the *Publish changes* button is displayed (6). Click on *Log* (7) to display a panel with the change log and validation result. Here you can view the changes made to the type and any errors. You can only publish your changes for the type individually after correcting errors (see Chapter 3.9).

4.1.1 Workflow Modeler



1 Toolbar

The toolbar contains the elements you use to build the structure of your workflow. See chapter 4.2 for a detailed element description with usage examples. You can also access the following tools for working with the elements in the canvas here:

- Hand tool
- Lasso tool
- Add/remove space

2 Canvas

You add the elements to the canvas to build your workflow.

3 Workflow Properties

This area shows the properties of the element you have selected. The screenshot above shows the workflow dialog where you activate and set the simplified view. For more information about the simplified view, see chapter 4.6.

4 Error Handling

Either the *Hide errors* or *Show errors* button is displayed in this area:

- Show current error messages: If you activate this view, the current formal errors in the workflow design are displayed. This setting is especially recommended when you are finishing the design. If you deactivate the view, no formal errors are displayed.

5 Zoom Buttons

These buttons are used to customize the view.

- ⊙: Center the view and reset the zoom to the default value.
- ⊕: Each click enlarges the view by zooming in.
- ⊖: Each click zooms out to reduce the view.

6 Action Buttons

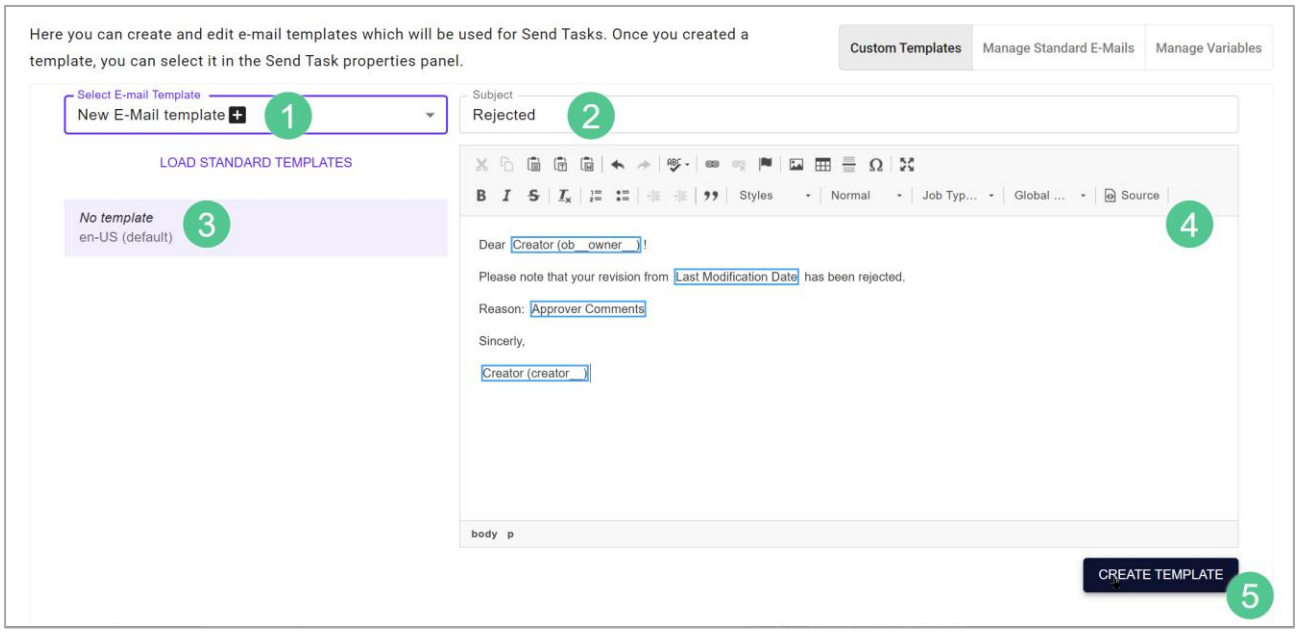
You can access the following actions in the bottom area:

- Delete diagram: This action deletes all the elements from the canvas.
- Undo: The last action is undone.

  Keyboard shortcuts: : CTRL + Z / CMD + Z
- Repeat: The last undone action will be restored.
  Keyboard shortcuts: CTRL + Y or SHIFT + CTRL + Z / CMD + Y or SHIFT + CMD + Z

4.1.2 Custom Templates

In the open process type, navigate to ☰ > *E-mail management* > *Custom templates*. Here you can create and edit the e-mail templates that are to be used later when sending tasks in processes.



1 Select E-Mail Template

When you create or edit a template, the language variants are displayed here. In the current version, you can create German and English language variants, whereby the version for US-EN must be created first.

2 Subject

In the selection list, select the e-mail template you want to edit, or select the *New e-mail template* entry to create a new template. Optional: If you click *Load Standard Templates* below the list, the templates stored in the system are loaded into the selection list. You can then also edit the default templates.

3 List of Language Variants

Enter the subject of the e-mail in the selected language variant.

4 Editor

Enter the content of the e-mail in the editor. You have numerous formatting options, and you can add system variables (see chapter 5.2) and job variables using the drop-down lists *Job Type Placeholders* and *Global Placeholders*. Variables of the following types are available:

- Single line and multiline input area (see chapter 5.1.14)
- Single-select and multi-select (see chapter 5.1.14)
- Date picker with/without time (see chapter 5.1.6)
- User (see chapter 5.1.2)
- Numbers (see chapter 5.1.12)
- Combo box (see chapter 5.1.4)
- Comfort Grid (see chapter 5.1.5)
- Advanced Grid (see chapter 5.1.2)

Other variable types cannot be inserted.

In the BPMN use case, only the following variables are populated:

- *JM_JOB_CREATOR_NAME*
- *JM_JOB_CREATOR_EMAIL*
- *JM_JOB_LIST_VARIABLES*
- *LINK_JM_JOB_DETAIL*

It is not possible to insert other types of variables in this case.

5 Create Template

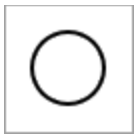
Click the button *Create Template* to save the created template. The text entered under *Subject* is used as the name.

4.2 Available Elements

Some concepts of the "Business Process Management Notation" (like Service Tasks, Script Tasks, Events, etc.) are not immediately understandable for casual users. Using this technology requires a comprehensive range of professional experience. If this is present, you will soon be rewarded and amazed at what is possible with it.

4.2.1 Start Event

The start event initiates a process instance. Each BPMN workflow requires exactly one start event.



Parameter

You can enter a name for the start event in the properties dialog.

4.2.2 End Event

The end event ends a process instance. Each BPMN workflow requires exactly one end event.

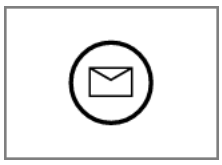
Parameter

You can enter a name for the end event in the properties dialog.

4.2.3 De-Archive Event

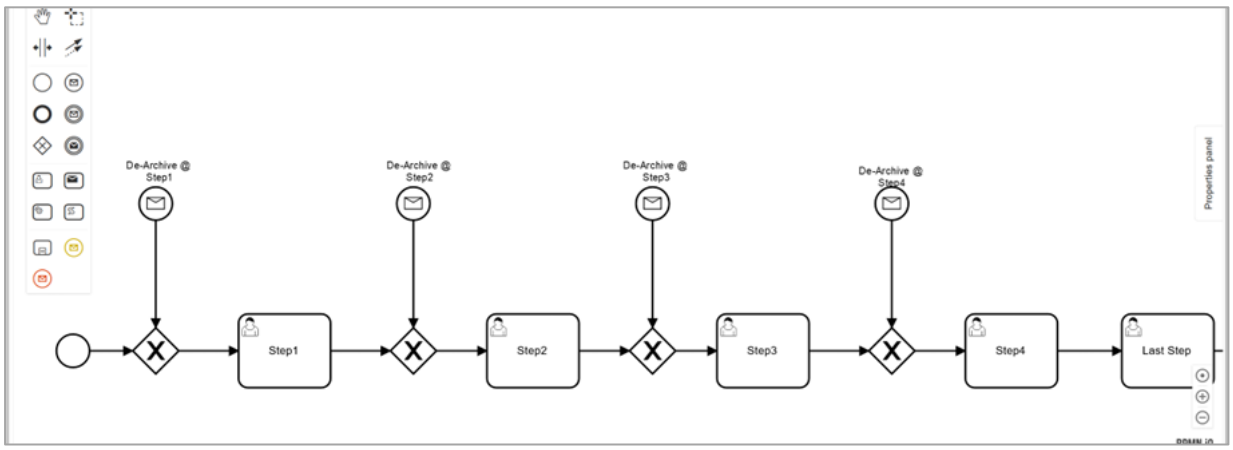
Occasionally it is necessary that terminated or canceled jobs can be reactivated with BPMN workflows. You can adapt your existing BPMN workflows accordingly. A standard BPMN message event of type *StartMessageEvent* is used for this purpose.

The De-archive event must be accessible to the StartMessageEvent. Appropriately prepared jobs can later be restored by users in the Job Manager. If the associated type has been deleted, processes that have already been canceled and terminated cannot be reactivated.



Implementation in the Workflow Modeler

The workflow can be started either by the regular Start event or a De-archive Event stored as De-Archive StartMessageEvent. You can theoretically insert a dearchiving event with a gateway before each workflow step in order to restore at this point at a later date.



> Notes
>
> - Currently, a BPMN job can only be reactivated if the entire workflow does not contain fewer steps at the time of reactivation than at the time of job creation.
> - Deleted jobs cannot be reactivated, even if they were previously properly terminated or canceled.
>
> In practice, not every entry point proves useful. If a job were reactivated only after an element such as *CreateNode*, the person reactivating the job would be missing the previously uncreated node in the current workflow step. Therefore, careful planning is required for each user scenario.

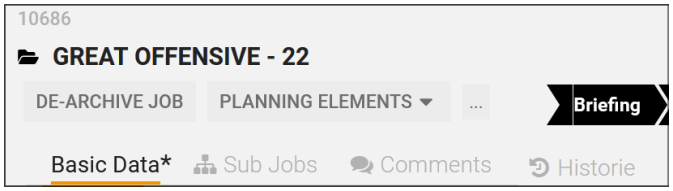
Job Manager users can reopen a completed or canceled job with BPMN workflow.

Prerequisite

- You have the permission *De-archive* in your Job Manager role.

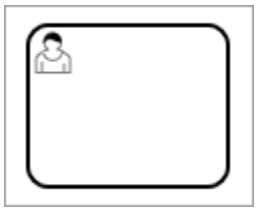
You will then find the button *De-archive Job* where the button *Close* is otherwise located. If several reactivation points exist, a selection dialog appears where the user can choose from a list to reactivate the job in a selectable workflow step.

User view in a datasheet with button to reactivate a canceled or terminated job with BPMN:



4.2.4 User Task

User tasks play an important role in modeling business processes that require human interaction. Each step in a process instance must be described as a task. The *User task* element is used to model the work to be performed in the system by a human actor. This can include filling out a form, reviewing documents, or approving an order.

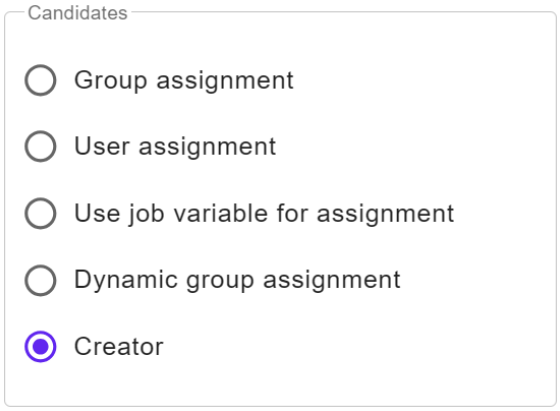


Name

For a user task, you have to enter a name in the *Properties panel*.

Assignment

As user tasks are not executed automatically, you must assign them to a group or individual. Alternatively, you can also select the responsible person using a variable on the datasheet (e.g., with the *Creator* variable).



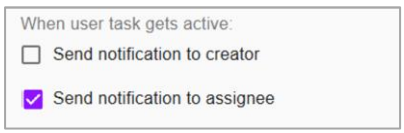
- *Group assignment*: When the user task is activated, it is displayed in the *Available tasks* list for the group. Every user who is part of the group can accept the task.
  With this setting, you select a user group in the pick-list.
- *User assignment*: When the user task is activated, it is assigned directly to the selected user.
  With this setting, you select a user in the pick-list. You can search through the users with access to the module based on logins, e-mail address, first name, and last name. An auto-complete helps you to select the user.

- Use variable for the assignment: When the user task is activated, it is assigned directly to the user who is entered in the selected variable.
- *Dynamic group assignment*: This assignment makes it possible to dynamically assign user tasks in a workflow to user groups depending on the scenario. To use this assignment, you must create and place the *user group* variable on the data sheet. This can be pre-assigned with a default entry. If the users are to select a group later, the *user group* variable must be created as a mandatory field. See chapter 5.1.18.
- *Creator*: The user task is assigned to the creator of the process.

> Note
>
> The default *Creator* variable is always available. To select a different variable for the assignment, you must first populate the datasheet layout with the desired variable.

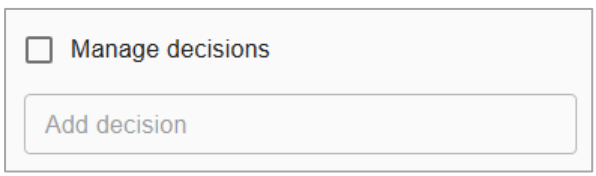
> Attention!
>
> If the variable is not filled out when the user task is activated, the process hangs and cannot be ended. Therefore, ensure that the variable is filled out, for example, by having the variable filled out as a mandatory field in an earlier step (see chapter 3.6.2).

Based on the assignment, you can control which notifications are sent when a user task is activated. If the task is assigned to a group, a notification is sent only to the creator. If a specific user has been defined as the processor, notifications can be sent to both the creator and the processor.



Decisions

To make it easier to create objects, decisions such as *Approve,* or *Reject* can be enabled and added directly in the properties of the user tasks. Decisions are displayed as buttons in the header of the datasheet. When the user clicks a button, the relevant sequence flow is performed.



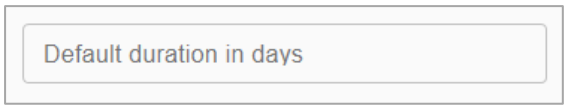
If you activate the *Manage decisions* checkbox, an exclusive gateway is added in the canvas after the user task. The *Properties panel* for the task displays the default decisions *Forward*, *Approve*, and *Reject*.

You can use these decisions, remove them (recycle bin icon) and enter additional decision options in the *Add decision* text box and save them by pressing ⏎.

You assign the decisions to the sequence flows by connecting the exclusive gateway to the following elements.

Default Duration

You can also define a default duration in days for user tasks. The default duration is used to calculate the due date for the task.



Notes Box

Enter your description in the *Notes* box. This will help you or other colleagues later to track your decisions and thoughts about the workflow structure.

Change Assignee in Workflow Step

The editor of the current workflow step can be changed without changing the job status. This option is only available if the user task in the BPMN workflow contains the *group assignment* or the *job variable* for the assignment.

It is not possible to change the editor if they are set as *user assignment* or *creator* in the *User Task Properties*. The new editor will be notified if the notification setting is

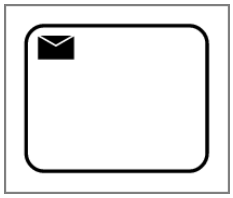
configured. To accomplish this, scroll down on the right side of the *User Task Properties* panel and activate the checkbox at *Send notification to editor*.

The user assignment of the workflow step can thus be set by editors in the ... menu of the job datasheet, but also in the *Details* area of an open job with the pencil icon in the *Editor* field.

> Note
>
> The *Assignee* field must not be empty. The new assignment can be selected only from a user group associated with the active user task, or from the user group corresponding to the user variable in use.

4.2.5 Sending Tasks

When a sending task becomes active in the workflow, the system sends an e-mail. The administrator selects a template for the e-mail in the properties dialog and specifies the recipient(s).

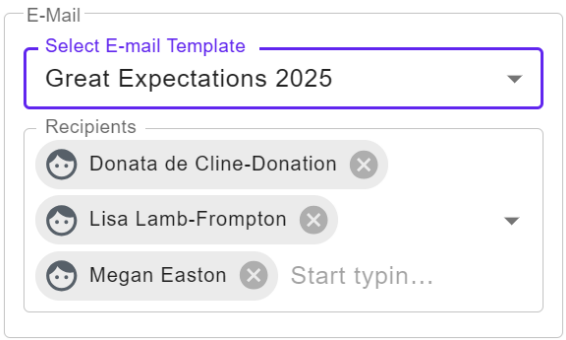


In the Workflow Manager, in the *Send Task Properties*, select *Participants* as recipients.

Later, when executed in this step, the send task will send a notification to all current job participants. All participants will also receive notifications under the bell icon in the menu bar.

E-Mail Template

You must create email templates to choose from in advance from on the *E-Mail Management* page on the *Custom Templates* tab, see chapter 4.1.2. How to create templates is explained in chapter 4.4.



Recipients

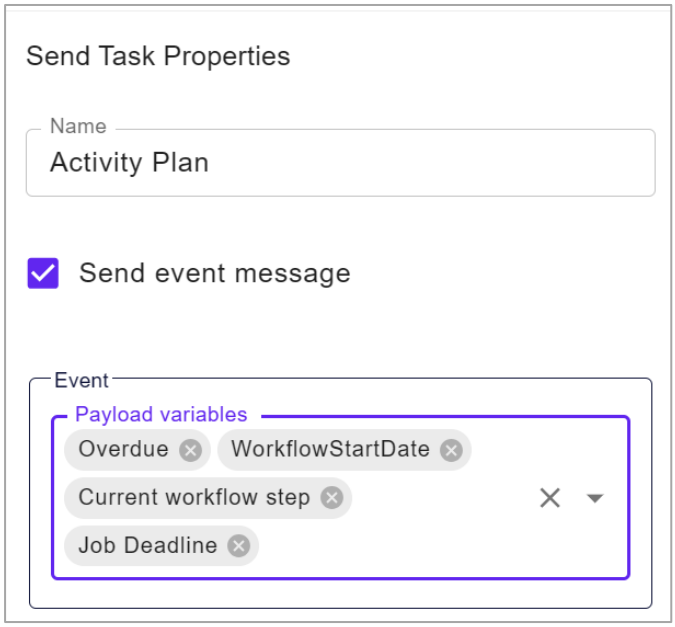
Select one or more recipients for the e-mail to be sent.

4.2.5.1 Send Event Message

Activate the *Send Event Message* checkbox to perform a synchronization with the selected payload variables in the direction of the *Activities* module.

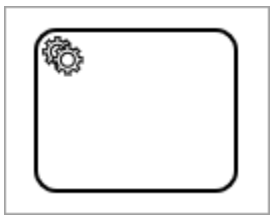
1. Create the *send event* in the Workflow Modeler or select an existing one.
2. In the Send Task Properties, activate the Send event message check box.
3. The letter icon in the BPMN *Send Task* changes its appearance to ⌘. All events that are synchronized with activities can be recognized by this icon in workflows.
4. In the *Event* field, select all the variables from the drop-down menu you want to send to the Activities module as payload variables for synchronization.

You have established the connection to the *Activities* module. Activities that are configured accordingly can be synchronized with workflows from the *Jobs* module that are based on this type.



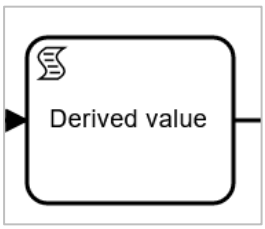
4.2.6 Service Tasks

> Note
>
> The service tasks are accessible in the interface from version 7.0. Note, however, that these service tasks can only be used to synchronize a job with a planning element in the *Marketing Planner* module. If you require any further assistance, please contact your Uptempo representative.
>
> All job element actions use the *JobRestService*.



4.2.7 Script Task

The JavaScript action, called a *script task* in BPMN, allows developers to add JavaScript code to process execution. Script tasks are used to perform minor calculations and application logic. Applications include data validation, data conversion, and integration with external systems to query or transfer data.



> Note
>
> Script tasks are intended for small-scale tasks. For more sophisticated or complex use cases, it is recommended to use either separate Service Tasks or Application Servers.

When a process instance calls up a script task, it creates a corresponding job and waits for its execution. When this job is completed, the process instance proceeds to the next logical step.

You can use this to quickly add additional custom logic to any business process. First, however, you need to familiarize yourself with the specifics of the script task.

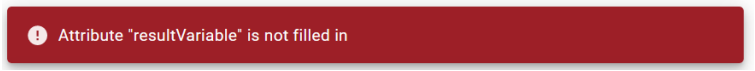
The basic structure of the script template is always static and therefore cannot be changed. Since the editor, apart from the coloring of the syntax, has no real debugging,

you can only rely on the reviewing of gross errors before the type is published. The final functional test is then only done with a job based on this process type.

```
1 function calculate(creator__) {
2   var result
3
4
5
6
7   return result
8 }
9 calculate(creator__);
```

You insert your code starting at line 3.

You can use multiple payload variables in a script task. You must select at least the result variable, the payload variable is optional. If the task does not contain the required return value (`resultVariable`), then you will receive an error message during



Example

You use the `creator__` variable as the payload and want to output the login name of the creator in a job variable at the end of the script task. Subsequently, you will see that more than just the login name is returned:

```
{"name":"Ron Swanson","id":1120,"login":"ron.swanson"}
```

This content must first be parsed. You can do this with the following code:

```
const json = creator__;

const obj = JSON.parse(json);

result = obj.login;
```



```
const myJSON = JSON.stringify({
  name: 'Ron Swanson',
  id: 1120,
  login: 'ron.swanson',
});
const obj = JSON.parse(myJSON);
var result = obj.login;
console.log(result);

CONSOLE ×

ron.swanson
```

You now actually get the login name alone as a return string.

```
ron.swanson
```

> Note
>
> Proceed carefully and note that you can also assign the script task *Result* to variables that should not be overwritten at all, such as IDs.

With the appropriate programming skills, you can define rules in JavaScript to describe which combinations of events and data lead to predefined results.

In addition to rules, you need to define what happens when the input matches multiple rules or no rules (error handling).

Error Handling

In addition to the rules, you must define what happens if input matches more than one rule or none (error handlers). Error handlers can be added to a script task according to the procedure for service tasks. To do this, press the *Add Error Event Handler* button in the opened *script task properties*.

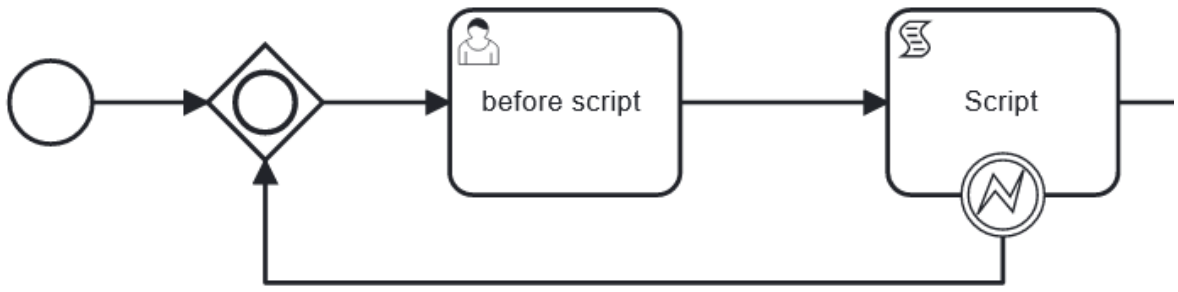


Table with examples of payload variable return values

Displayed payload variable / Unique name	**Unparsed payload after query by script task**
Assignee: `ctx__assignee`	`{ name=Elena Employee, id=1119, login=elena.employee }`
Job-ID: `job__id`	`437`
Description: `description__`	`"String"`
Creator: `creator__`	`{ "name":"Ron Swanson", "id":1120, "login":"ron.swanson" }`
Job status:-`job__state`	`ACTIVE`
Job type: `job__type__pseudo__variable`	`7877`
Current workflow step: `workflow__timing`	`{duration=null, dueDate=null, startDate=null}`
Job name: `job__name`	`Disney Summer Campaign 2023`
Workflow: `workflow__object__id`	`2829`

Return Value of the Script

Here you can specify a variable on the datasheet as target. This serves as a buffer for the result of the script task and can be evaluated elsewhere. It is sometimes necessary to create separate fields for this on the datasheet layout, perhaps hidden on their own tab. This way, users are not confused and are not tempted to briefly fill in such a standard field. These variables serve as a buffer for the `result` of the script task, which can then be evaluated later and used as a pre-filled parameter.

Convert numerical value to string

You have performed a calculation and want to write the result into a text variable. Only line 1 alone does nothing. The result is not written into the variable. It is only through the conversion in line 2 that the result now appears in the filled text variable in the next process step.

```
1    var num = (23*11*65);

2    var result = num.toString();
```

Convert string to floating-point number

The variables for Total amount of purchase and Shipping costs are created as text fields (Single inputline) in the datasheet, but contain numerical values. The conversion of the variable is quite simple.

```
1    var price = parseFloat(system__price);
```

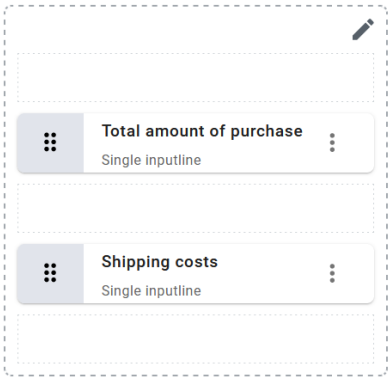
This allows you to perform calculations.

Note

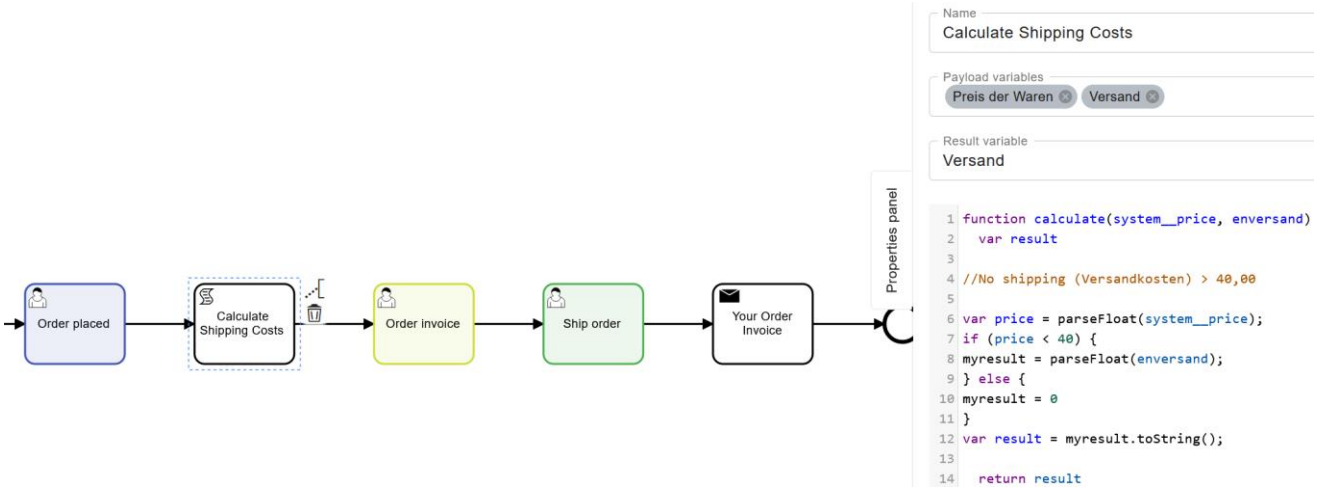
It is worth noting that only the last value declared in the script is returned. The content of `result` can only be assigned to one variable at a time. If you have any questions, please contact your Uptempo contact person.

4.2.7.1 Process with a Script Task

Our example workflow is to determine whether shipping costs are incurred for the value of goods in an order. The calculation is done via Script Task.



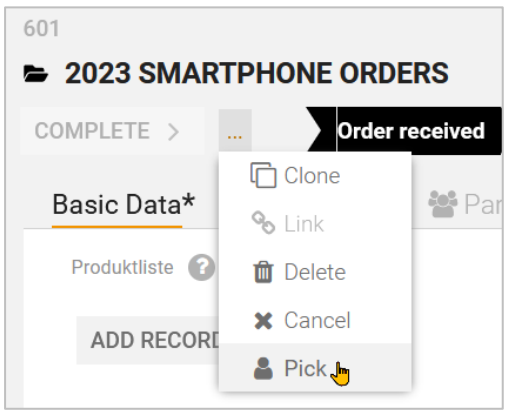
The default shipping costs are prefilled in the datasheet in the variable `Shipping costs`. Whether an invoice is issued without shipping costs is not difficult to calculate.



The Payload Variables (`Total amount of purchase, shipping costs`) are text variables and their content must be converted to a floating-point number. The condition is that no shipping costs are calculated starting from a goods value of €40.00. How much the shipping costs are below the limit is predefined in the `Shipping` variable. The script task evaluates the price of the goods from the order in the previous step, the condition for shipping costs is checked in an `if/else` statement. If there are none, the `Shipping Costs` variable is set to *0*.

```
//No shipping (Shipping costs) > 40,00

var price = parseFloat(system__price);

if (price < 40) {

myresult = parseFloat(enversand);

} else {

myresult = 0

}

var result = myresult.toString();
```

An e-mail with the calculated shipping costs is sent as a preliminary invoice at the end of this application example.



> Note
>
> For workflow steps that are not permanently assigned to a person, but must be actively accepted by the editor, the value calculated by the script task in the previous intermediate step is only updated in the datasheet after the step has been accepted for editing. (Menu command: *Pick*).

4.2.8 Intermediate Event Throw Message

There are two types of intermediate events. An intermediate event in the sequence flow can either trigger an event or respond to the receipt of an event. Intermediate events in the sequence flow have exactly one incoming and one outgoing sequence flow.

Intermediate event messages are used to send messages from one process to another process and trigger an action there. After the event is triggered, the process continues.

4.2.9 Intermediate Event Catch Message

An intermediate event catch message is used in processes to catch messages from a parent process and trigger appropriate actions. The sequence is paused until the event occurs.

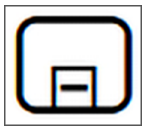
For example, the flow is interrupted until a message is received. The process continues after the event has occurred.

4.2.10 Sub-Actions

Sub-actions are actions independent of the central BPMN workflow that are executed in parallel. Event-based subprocesses in the Workflow Modeler are identified by a dotted outline.

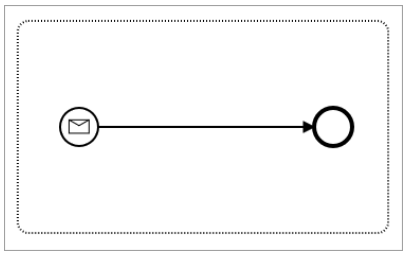
The event used to trigger one of these subprocesses is configured via a start event.

Create extended subprocess



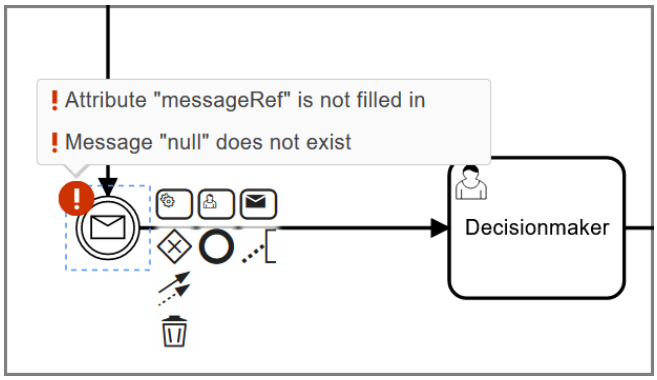
The extended subprocess always looks as shown below and always starts with a message start event. For example, such a process responds to MAPL update events.

Message events are used to send and receive messages between two separate processes. Message events serve as a mechanism for communication between processes.



After you created the subprocess, add an intermediate event catch message to the main process. The Workflow Modeler still reports errors at first as necessary attributes and referencing are missing.

After you create the subprocess, add an intermediate event catch message to the main process. The Workflow Modeler still reports errors at first. Necessary attributes and referencing are missing.



To resolve this issue, highlight the placed intermediate catch event message symbol and select the option *Mapl Update Event* in *the Intermediate Catch Event Properties* panel in the *Message* field.

This fixes the errors.

The following two workflow elements can be used to configure which actions are initiated after the deletion or cancellation of an associated job.

Message events are used for these use cases.

Create Subprocess „Cancel Message“

You create an event subprocess that responds to a *Cancel Message* type event. Example: A user has cancelled a job prematurely. You can specify what should happen when the job is cancelled.

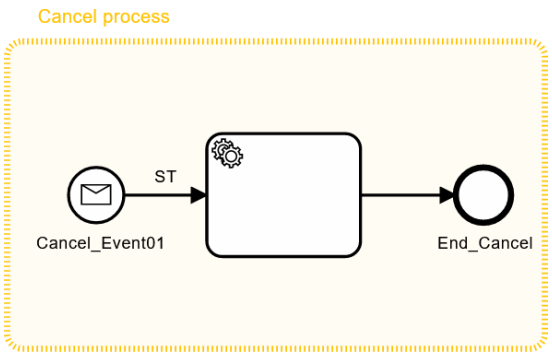
You can:

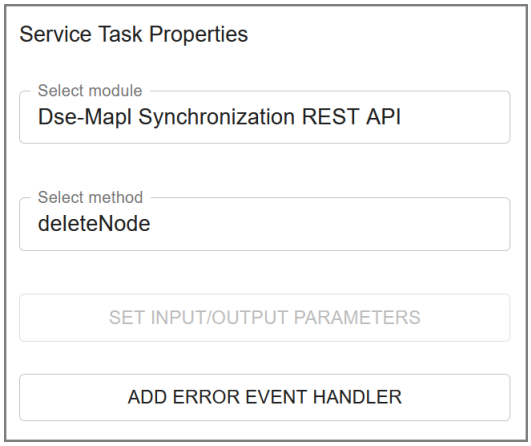
- Delete synchronized planning elements
- Cancel sub-processes

The planning element in Planner is to be removed by the event.

The following methods for the *MaplNodeSyncRestService* are available:

- *deleteNode*
- *createNode*
- *moveNode*
- *updateNode*





Adjust the properties in the service task of the sub-process accordingly. The method *deleteNode* is needed to delete the planning element in Planner on termination.

1. In the drop-down menu, select the module: *Dse-Mapl Synchronization REST API.*
2. From the drop-down menu, select *deleteNode*.
3. After you create and configure the subprocess, add an *Intermediate Event Catch Message* to the main process.

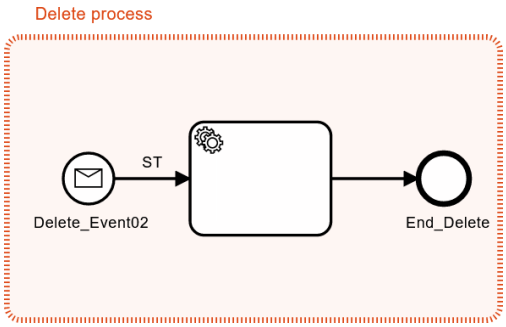
You have defined what happens if a user cancels the job prematurely. The planning element is deleted when the event occurs in the *Cancel Message* sub-process.

Create Subprocess “Delete Message”

You create an event subprocess that responds to an event of type *Delete Message*. Example: A user has deleted a running job.

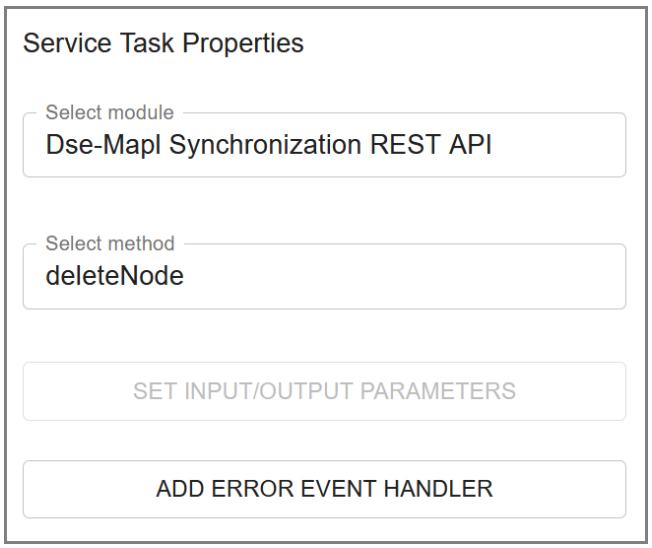
You can specify what happens after the job is deleted. For example, if the associated node in Planner is also to be deleted, you must create a corresponding service task.

Adjust the properties in the service task of the sub-process accordingly. Here, too, the method *deleteNode* is used to delete the planning element in Planner in the same way when it is cancelled.



Adjust the properties in the service task of the sub-process accordingly.

1. In the drop-down menu, select the module: *Dse-Mapl Synchronization REST API.*
2. From the drop-down menu, select *deleteNode*.
3. After you create and configure the subprocess, add an Intermediate Event Catch Message to the main process.



After publishing the changes, this job will be synchronized with the related elements in Planner. The planning element in the Planner is removed by an event with the above settings for the two subprocesses presented.

It is also possible to process several service tasks one after the other in one subprocess. In this way, it is also possible to delete sub-jobs as part of a subprocess.

4.2.11 Gateways

Gateways are used to control the process sequence: If the condition is true, the process takes the appropriate course. The available gateways are described below. For information about setting up conditions with gateways, see chapter 4.5.

Exclusive (OR)

All the conditions are evaluated, but only one condition can be true; therefore, only one sequence flow is performed. The conditions can be established using user decisions or *Number* or *Single select*-type variables.

Inclusive (AND)

All the conditions are evaluated, and anything from one condition to all the conditions can be true; each sequence flow with a true condition is performed. The conditions can be established using *Multiselect*-type variables.

Parallel

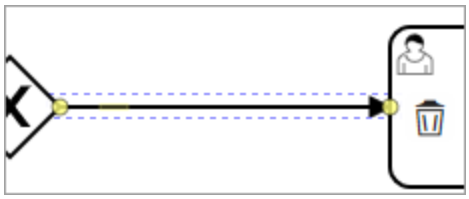
All the subsequent sequences are performed; conditions are therefore not required.

Parameters

You can enter a name for the gateways in the *Properties panel*. Other settings are based on the conditions that you establish. For more information, see chapter 4.5.

4.2.12 Sequence Flow

A sequence flow connects tasks, gateways, and events and creates the process flow between the start and end element. Conditions for decisions at gateways are also described on the sequence flows.



Default Sequence Flow

For sequence flows departing from a gateway, you have the option to specify a default sequence flow. The default sequence flow is executed if none of the other conditions are met.
How to set a default sequence flow is described in Chapter 4.3.2.

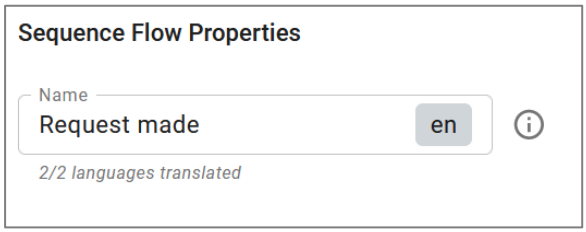
Parameters

In the case of sequence flows that connect exclusive and inclusive gateways with subsequent elements, you define the decisions in the *Properties panel*.

Customizable Button Name

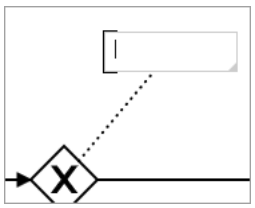
You can customize the button names that appear in the open job datasheet in the Workflow Modeler. The name entered under *Sequence Flow Properties* will later be

used as the button name. If no name is stored, the default name from the text resources appears.



4.2.13 Comments

You can use comments to add explanatory details about the elements in the process.



Parameters

You can also enter and modify comments in the properties dialog.

4.3 Creating a BPMN Workflow

This chapter describes how you create BPMN workflows using the Workflow Modeler.

> Attention!
>
> You have to fix all formal errors before you can publish a newly created or modified BPMN workflow. Publishing with formal errors can permanently damage the job type.

4.3.1 Adding Elements

You can add a start or end event, a user task, or a gateway in two ways:

Toolbar

1. Click the relevant icon in the toolbar.
2. Click the point in the canvas at which you want to add the element.

The element is added.

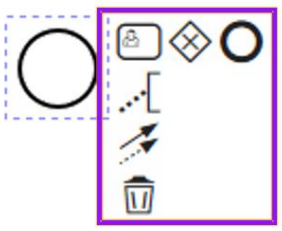
Adding Related Elements

You use this option when you wish to add elements that are directly related to one another.

Prerequisite: You have already added one element. This element must be before the element to be added in the process sequence.

1. Click the added element after which you wish to add a related element.

   A toolbar is displayed to the right of the element:



2. In the toolbox, click the symbol for the next element.

The selected element is added. It is connected to the previous element through a sequence flow.

4.3.2 Connecting Elements

Once you have positioned elements, you must connect them using sequence flows. Please refer to the following sections, which explain how to connect using the toolbar or using elements that have already been inserted. In addition, you can define a default sequence flow for sequence flows that leave from a gateway, see the last section.

Toolbox

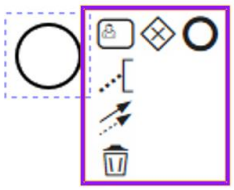
1. In the toolbox, click the symbol for sequence flows.
2. In the canvas, click the element at which the sequence flow starts.
3. In the canvas, click the element at which the sequence flow ends.

The sequence flow connects the elements.

Elements that Have Been Added Already

1. Click an element that you have already added to the canvas area.

   A toolbox is displayed to the right of the element:



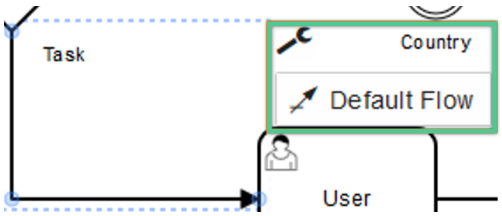
2. Click the sequence flow symbol.
3. Click the element to which you want to connect the starting element.

The sequence flow connects the elements.

Set Default Sequence Flow

> Note
>
> You can only set a default sequence flow that originates from a gateway.

1. Click the sequence flow you wish to set as default.
2. The icon is displayed on the frame of the sequence flow.
3. Click > > Default flow.



You have set the sequence flow as the default sequence flow.

4.3.3 Positioning Elements

To display all elements in a clear way, you can reposition individual elements of the diagram.

> Note
>
> If you move connected elements, the connections are not removed; instead, they are updated and adapted to the desired position.

Positioning an Individual Element

1. Click the element and hold the mouse button.
2. Drag the element to the desired position and release the mouse button.
3. Alternatively, you can fine-position a selected element or group using the keyboard with the four direction keys or move it in larger steps with SHIFT + direction key.

You have positioned the element.

Positioning Multiple Elements

1. Hold down CTRL + SHIFT / CMD + SHIFT and click the elements you want to move together.

   The elements are highlighted together.
2. Click one of the highlighted elements and hold down the mouse button.
3. Drag the elements to the desired position and release the mouse button.

You have positioned the elements.

Positioning an Area

If you wish to position an area of the diagram containing several elements, proceed as follows.

1. In the toolbox, click the symbol.

2. Move the mouse cursor over a free area of the canvas and hold down the mouse button to select the area that you wish to position.
3. Click one of the highlighted elements and hold down the mouse button.
4. Drag the selected elements to the desired position and release the mouse button.

You have positioned the elements.

Adding/Removing Space

1. In the toolbox, click the symbol.
2. Move the mouse cursor over a free area of the canvas and hold down the mouse button to select the area that you wish to position.
3. Move the cursor horizontally or vertically.

> Note
>
> Depending on the use of the function, you can remove or add space only in one direction.

The space is added or removed.

4.3.4 Editing Elements

To edit elements, you open the Properties dialog box and edit the properties there. To do so, click an element. The Properties dialog box is then displayed to the right of the canvas. You may have to expand the dialog box.

Editing Multiple User Tasks

You can select multiple user tasks at the same time and assign them to a user or user group for editing.

1. Hold down the CTRL / CMD key and click the user tasks that you want to edit at the same time.

   You have selected the user tasks. The fields for the assignment are displayed in the Properties dialog box.
2. Assign the tasks. For information about performing the assignment, see chapter 4.2.4.

You have assigned multiple user tasks simultaneously.

Changing the Gateway Type

> Note
>
> If you have created conditions for a gateway, the options for changing the gateway type may be limited.

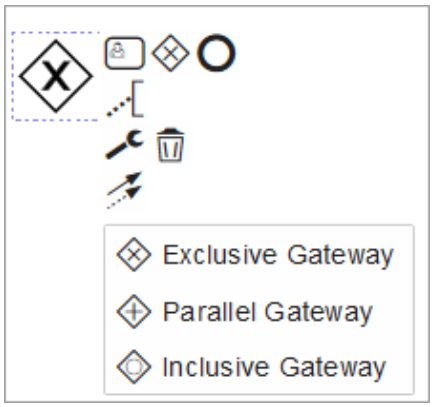
When you add a gateway, you always add an exclusive gateway to start with. To get a parallel or inclusive gateway, you have to change the gateway type.

1. Click the gateway whose type you want to change.

   A toolbar is displayed to the right of the element.

2. Click the symbol.

   A pick list is displayed:



3. Select the type.

You have changed the gateway type.

4.3.5 Editing a View

You can adjust the image section with the three zoom buttons above the BPMN.iO logo. Or you can use the following key combinations to adjust the zoom factor:

Windows/Mac key combination	Function
CTRL + 0 / CMD + 0	Reset zoom factor to default
CTRL + + / CMD + +	Zoom in
CTRL + - / CTRL + -	Zoom out
SHIFT + Scroll wheel	Scroll horizontally

Trackpad owners can use the pinch-zoom gesture to adjust the area with two fingers by moving them apart and bringing them together.

The zoom functions support the mouse scroll wheel. The current position of the mouse cursor over the workflow is considered. If you use such a mouse, hold down the CTRL / CMD key and scroll up to zoom in. Scroll in the opposite direction to zoom out.

Hold down the SHIFT key and use the scroll wheel to move the image section to the right and left. Without pressing another key, you can scroll up and down the visible area. Press the esc key to exit the selected BPMN tool. You can now click and hold directly in the white area of the workflow modeler until the mouse cursor turns into a hand, and freely move the image section. This method saves the detour through the toolbar. You do not need to select the hand tool separately.

Global Keyboard Shortcuts for the Workflow Modeler

Windows/Mac key combination	Function
CTRL + Z / CMD + Z	Undo the last action
CTRL + Y / CMD + Y or SHIFT + CTRL + Z / SHIFT + CMD + Z	Redo the state as it was before pressing STRG + Z / CMD + Z
CTRL + A	Select all elements
E	Quickly switch to text mode to enter a name for the selected workflow item
H	Deselect the current tool and switch to the hand tool
L	Lasso tool
S	Space tool

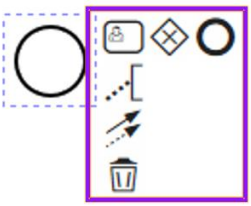
4.3.6 Deleting Elements

> Note
>
> If you have rejected decisions for a gateway, the gateway can no longer be deleted. To delete the gateway, you must first remove the subsequent user tasks.

Deleting an Individual Element

1. Click the element.

   A toolbox is displayed to the right of the element:



2. You can optionally click on the trash icon.
3. Or press the Del key or Backspace key on your keyboard to delete.

The element is deleted.

Deleting Multiple Elements

1. Hold down the CTRL/ CMD keys and click the items you want to delete.
2. Press the Del or Backspace key.

The selected elements are deleted.

Deleting the Elements in an Area

1. In the toolbox, click the symbol.
2. Click the canvas and hold down the mouse button to select the area that you wish to delete.
3. Press the Del or Backspace key.

The elements inside the selected area are deleted.

Deleting all Elements

1. In the Workflow Modeler, click the *Delete diagram* button in the bottom-right corner.

All elements are deleted. You must completely set up the whole workflow again.

4.4 E-Mail Templates

4.4.1 Creating an E-Mail Template

1. In the ☰ menu of the datasheet editor, select *E-Mail Management* > *Custom Templates*.
2. Enter a subject.
3. The subject is used as the name for the English version and template.
4. Enter the content for the English variant.
5. Click *Create template*.

   The English variant is created.
6. Click on the German variant on the left.
7. Enter the German subject.
8. Enter the content for the German variant.
9. Click *Create template*.

You have created the e-mail template.

4.4.2 Editing an E-Mail Template

1. In the ☰ menu of the datasheet editor, select E-Mail Management > Custom Template E-Mail Management > Custom Templates.
2. Select the template you want to edit.
3. Select the language variant you wish to edit.
4. Optional: Edit the subject.
5. Optional: Edit the content of the template.
6. Click *Save Changes*.
7. Optional: Repeat steps 3 to 6 for other language variants.

You have edited the e-mail template.

4.4.3 Deleting an E-Mail Template

For technical reasons, you can only delete the German variant and existing other languages of an e-mail template. The en-US variant is the default and therefore has no Recycle Bin icon.

1. In the ☰ menu of the datasheet editor, select E-Mail Management > Custom Template E-Mail Management > Custom Templates.
2. Select the template whose German variant you want to delete.
3. Click the recycle bin icon for the German variant.

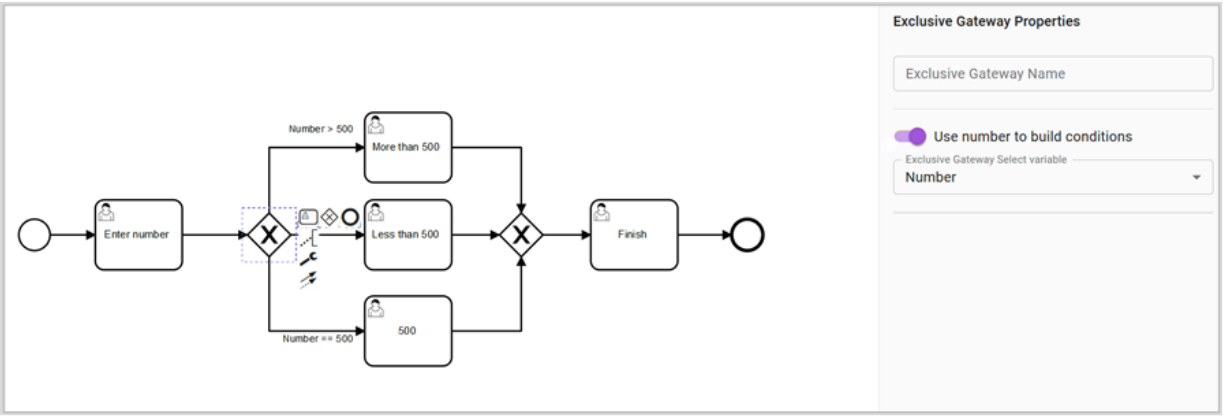
You have deleted the German version of the e-mail template.

4.5 Setting up Conditions

This chapter explains how to establish conditions using exclusive or inclusive gateways.

4.5.1 Exclusive Gateway: Using a Number

You wish to establish a condition in which a number is evaluated, such as:



Prerequisites

- You have assigned a number variable to the type (see chapter 5.1.12).
- You have added an exclusive gateway to the workflow.

Setting up a Condition

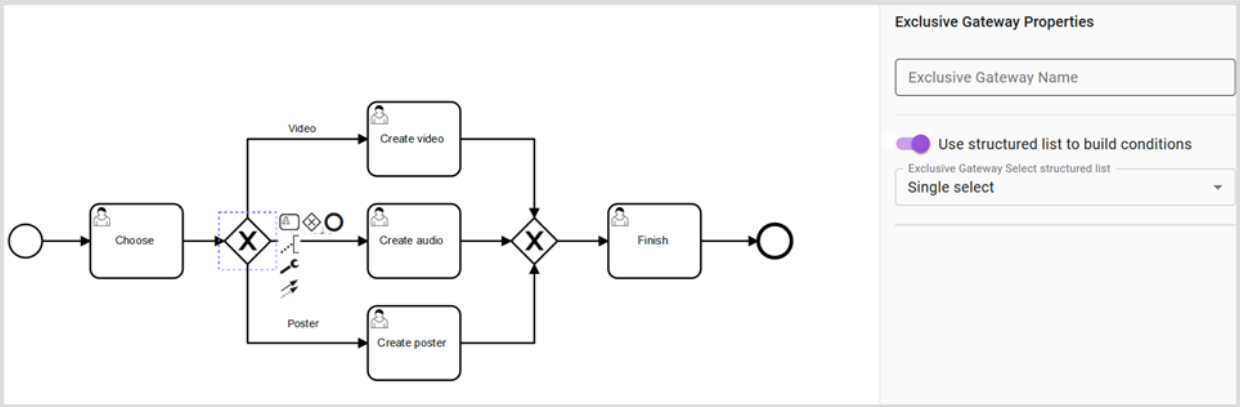
1. Click the gateway.
2. In the Exclusive Gateway *Properties panel* on the right, activate the function *Use number to build conditions*.
3. In the pick-list below that, select the variable that you want to use to create the condition.
4. Select all the outgoing sequence flows one after the other and enter the operator and value of the number variables in the *Properties panel*.

You have set up the condition.

> Note
>
> Negative numbers are permitted. If the number ranges for the various sequence flows overlap, the Workflow Modeler reports an error.

4.5.2 Exclusive Gateway: Using a Custom Structure

You want to establish a condition in which the selection is evaluated in a list. In this case, only one path is permitted to be executed.

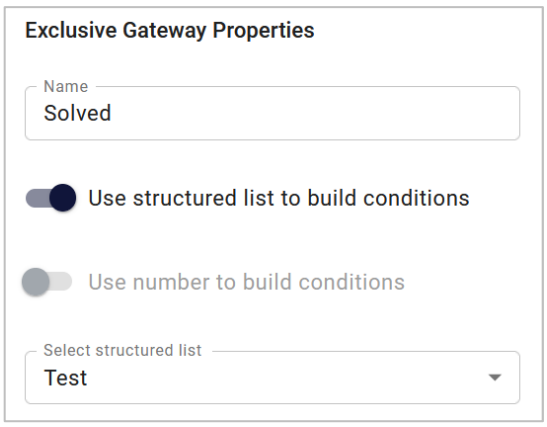


Prerequisites

- You have assigned a single select variable to the type (see chapter 5.1.15). A custom structure is assigned to the variable.
- You have added an exclusive gateway to the workflow.

Setting up a Condition

1. Click the gateway.
2. In the Exclusive Gateway Properties panel on the right, activate the function *Use structured list to build conditions*.



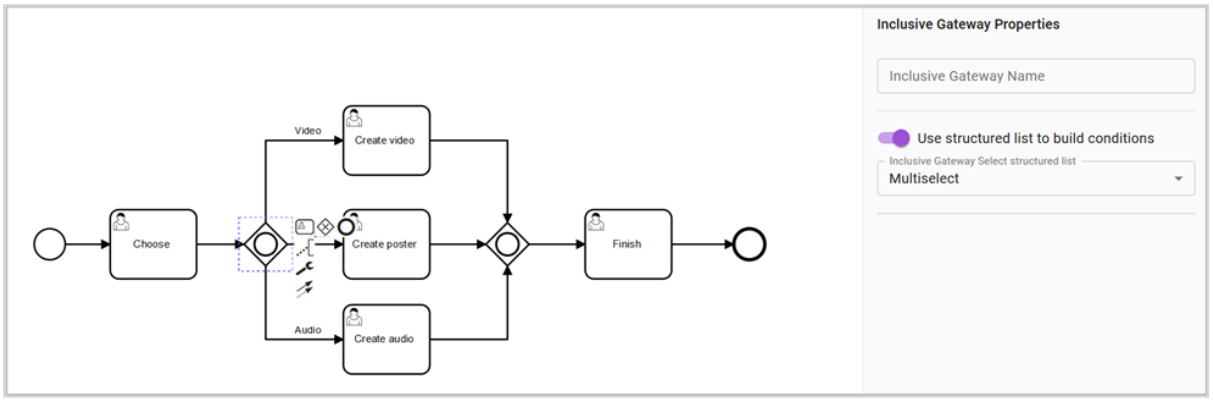
3. In the pick-list below that, select the variable that you want to use to create the condition. In our example above, the selected variable is called *Test*.
4. Select all the outgoing sequence flows one after the other and enter the list value with which the applicable path is to be executed in the *Properties panel*.

You have set up the condition.

> Note
>
> If not all the list values are assigned to a sequence flow, the Workflow Modeler reports an error.

4.5.3 Inclusive Gateway: Using a Custom Structure

You want to establish a condition in which the selection is evaluated in a list. In this case, multiple paths are allowed to be executed.

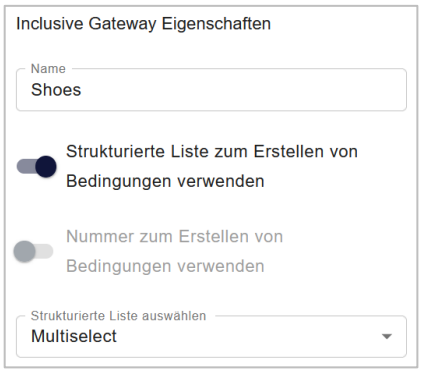


Prerequisites

- You have assigned a multiselect variable to the type (see chapter 5.1.15).
  A custom structure is assigned to the variable.
- You have added an inclusive gateway to the workflow.

Setting up a Condition

1. Click the gateway.
2. In the Properties dialog box, activate the function *Use structured list to build conditions*.



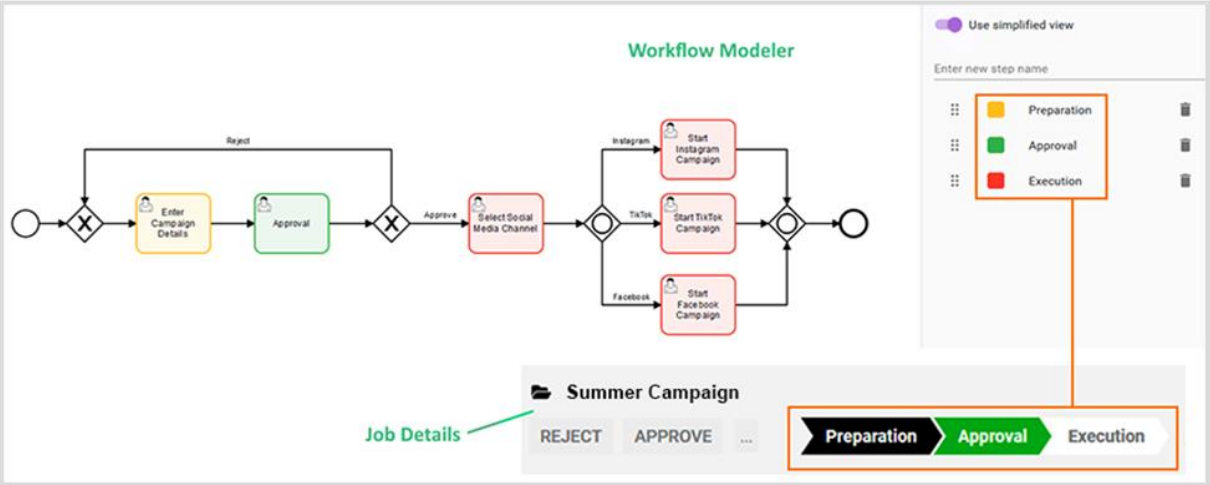
3. In the pick list below that, select the variable that you want to use to create the condition. In our example above, the selected variable is called *Multiselect*.
4. Select all the outgoing sequence flows one after the other and enter at least one list value with which the applicable path is to be executed in the Properties dialog box.

> Note
>
> If not all the list values are assigned to a sequence flow, the Workflow Modeler reports an error.

You have set up the condition.

4.6 Simplified View

You can use the simplified view to display the status of the job or data object in the workflow on the datasheet. In this case, you assign one or more user tasks to a process step. These process steps are displayed sequentially in the datasheet of a job or data object and show the progress made in the workflow:



> Note
>
> Processes are shown in the module's Kanban representation in read-only mode. This means that they cannot be dragged and dropped into another workflow step. The datasheet, on the other hand, can also be opened and edited in the Kanban view.

Prerequisite

- You have added at least one user task on the canvas.

Configuring the Simplified View

1. Click on a free space in the canvas.

   The *Properties panel* for the workflow is displayed.

2. Activate the *Use simplified view* checkbox.
3. Enter a name for the workflow step in the *Enter new step name* field.
4. Press ⏎ or the Enter key.

   The step has been created.

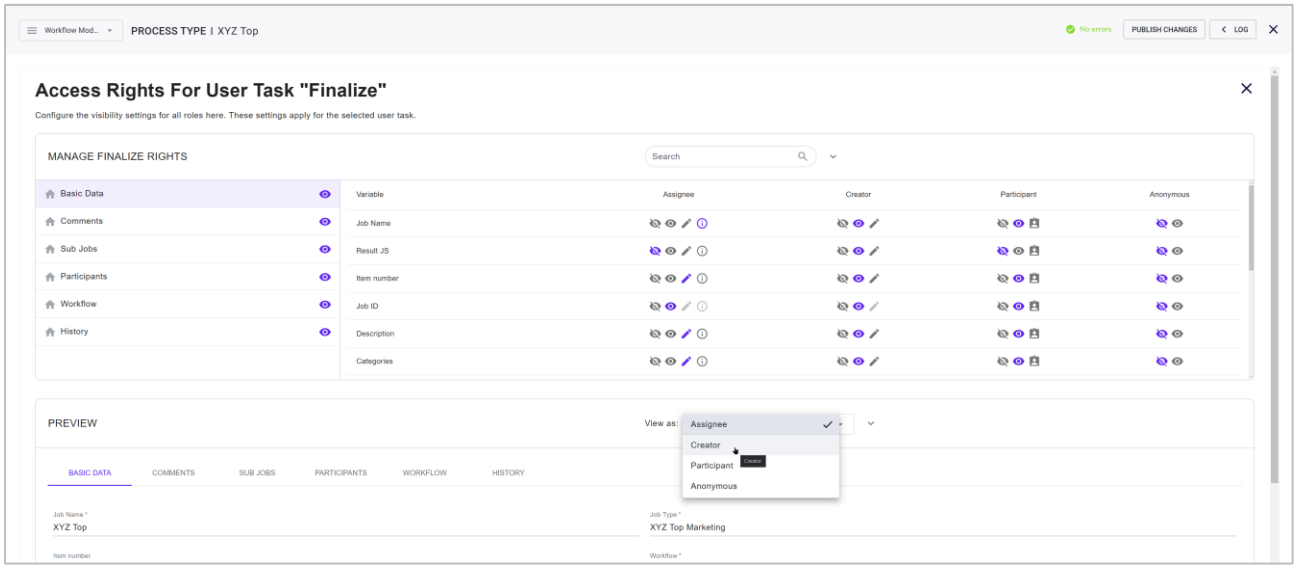
5. Optional: Click the colored field next to the step name and select a different color.
6. Select the tasks that you want to assign to the step.
7. In the *Properties* panel, select the step in the *Select step from simplified view* pick-list.
8. Repeat steps 3 to 6 until all the required workflow steps are created and all the user tasks are assigned to a step.

You have set up the simplified view.

> **Note**
>
> When you use the simplified view, the Workflow Modeler checks to make sure all the work steps are assigned to this view. If not all user tasks have been assigned, an error is displayed.

4.7 Configuring Access Rights

You define the visibilities of tabs, the datasheet and single variables in the editor for the process.



1. Open a process type.
2. Click the ☰ selection menu and select *Workflow Modeler*.
3. Go to the page and click in an empty area to select the entire workflow. You can also select a single user task.
4. Click on the user task whose visibility you want to set.

   The *Access rights* button appears.
5. Click *Access rights*.

   The page *Access Rights for...* opens.
6. Depending on what you activate, you either set up the visibility for the overall workflow or the corresponding user task.
7. At the top of the dialog, first select the user category for which you want to set the visibility.
8. Optional: Now edit the visibility of the datasheet tabs.
9. Optional: For visible tabs, you can specify whether variables are visible, editable, or mandatory.

   Mandatory fields to be completed are marked with * in the data sheet. Access rights can also be set for system variables.
10. Tip: If you are editing a process with many job variables, you can search for variables above the list of variables.

11. Optional: For the user task, identify the tab in the data sheet that you want to set as the default tab instead of *Basic Data*.
12. Click on the gray house icon next to the name of the tab.

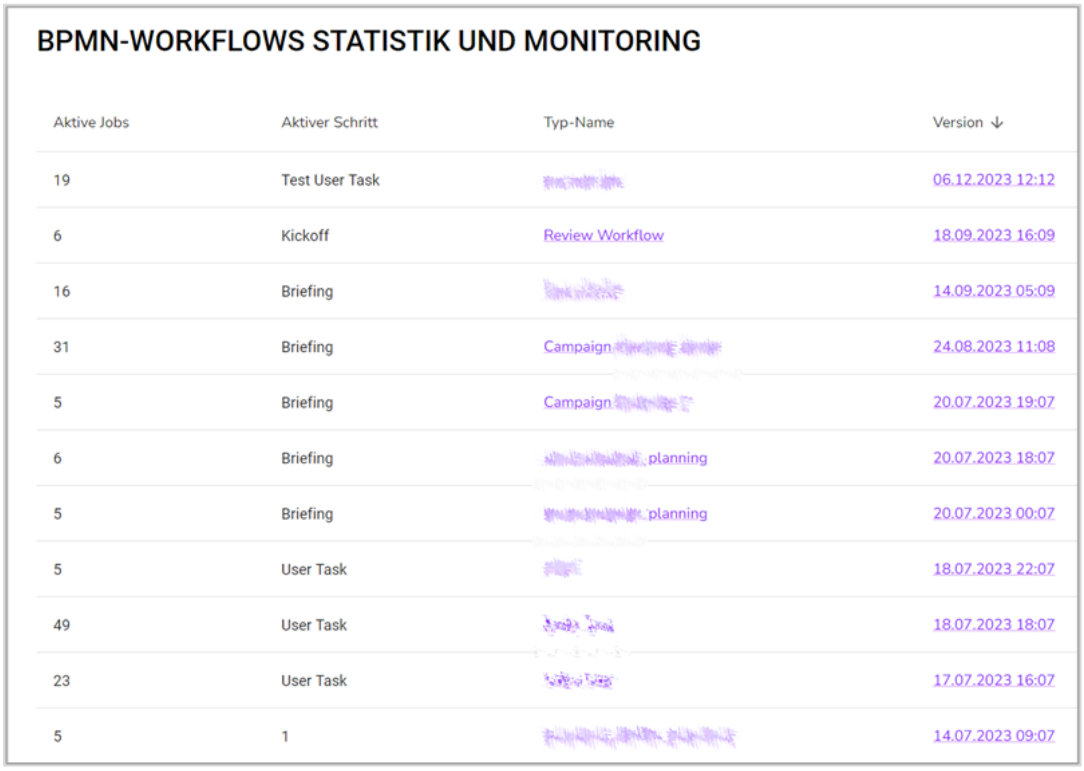
    The house symbol for the new default tab now appears in black.

    Changes to access rights are automatically saved when the page is closed.
13. Optional: Open the job to ensure that the selected tab is displayed as the default.
14. Repeat steps 7-14 if you also want to customize other user tasks.
15. You have changed the access rights and default tabs for the user tasks in the BPMN workflow.

> Note
>
> The default tab cannot be customized for the entire workflow, but only for individual user tasks.

Monitoring

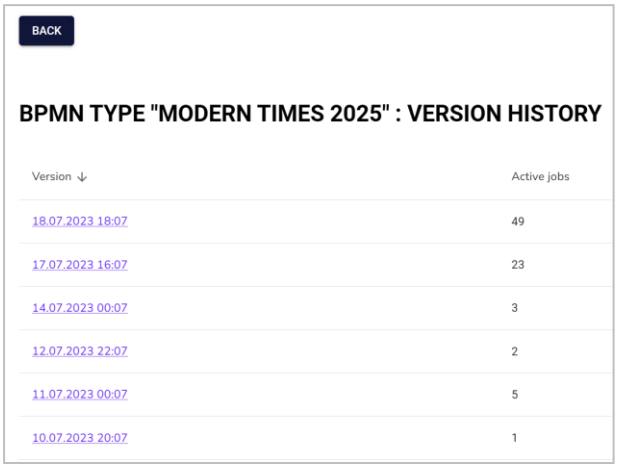
Administrators can access an overview of the running BPMN processes under > *Administration* > *Overview* > *Datasheet Engine* > *BPMN Monitoring*.



BPMN-WORKFLOWS STATISTIK UND MONITORING

Aktive Jobs	Aktiver Schritt	Typ-Name	Version ↓
19	Test User Task		06.12.2023 12:12
6	Kickoff	Review Workflow	18.09.2023 16:09
16	Briefing		14.09.2023 05:09
31	Briefing	Campaign	24.08.2023 11:08
5	Briefing	Campaign	20.07.2023 19:07
6	Briefing	planning	20.07.2023 18:07
5	Briefing	planning	20.07.2023 00:07
5	User Task		18.07.2023 22:07
49	User Task		18.07.2023 18:07
23	User Task		17.07.2023 16:07
5	1		14.07.2023 09:07

The table lists the ongoing processes by amount and active step. It also lists the name and version of the type. When you click the type name, the version history of the workflow opens:

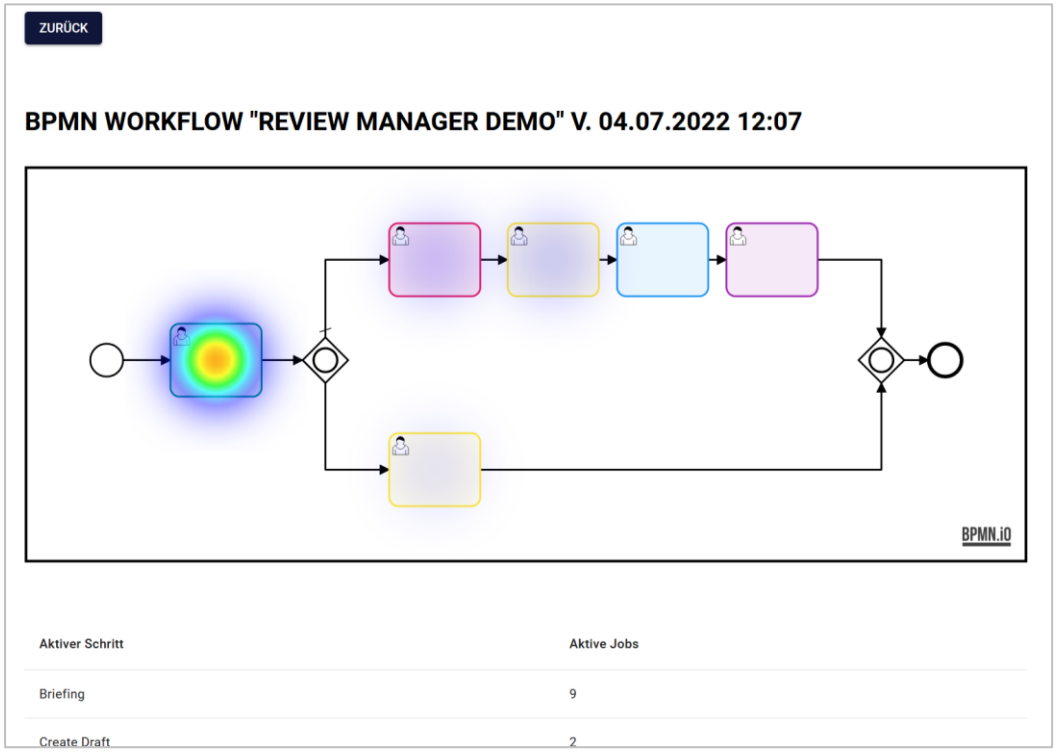


BACK

BPMN TYPE "MODERN TIMES 2025" : VERSION HISTORY

Version ↓	Active jobs
18.07.2023 18:07	49
17.07.2023 16:07	23
14.07.2023 00:07	3
12.07.2023 22:07	2
11.07.2023 00:07	5
10.07.2023 20:07	1

When you click on the link in the *Version* column, the workflow is displayed with a superimposed thermal image:



The more intense the color saturation of the hotspots, the more processes are in the respective process step. Steps without an active process are not colored.

5 Available Variables

Different variable types are used to allow you to group together and output data and information on a datasheet. The variables are placed on a datasheet using drag and drop.

There are two different types of variables:

- System variables that allow you to use the basic functions, for example, for displaying the creator or the current workflow step. When you create a type, system variables are created and – if absolutely necessary – placed on the datasheet automatically.
- Custom variables (text fields, selection fields, or an asset selector, for example) can be created as required.

Note: When you create a variable, fields that are mandatory are flagged with a * on the interface. Pre-filling means that users see already filled values in the datasheet but are allowed to overwrite the content.

5.1 Custom Variables

5.1.1 Action Button

With the variable named *Action button,* you integrate a button on a datasheet that opens a URL. The processor of the datasheet can use the button to access programming that is generally customer-specific.

Use the button *Close* at the bottom of the page or click in the Close box ✖ at the top right to navigate back to Job Manager.

Name	Description
Unique name, technical name	Note: This is visible only if the variable is changed. None of the fields are visible when you create the variable. For details, see chapter 2.2.
Display Name	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. c
URL	URL that is opened when you click the button.

Name	Description
Help text	Define the help text that can be displayed to users.

5.1.2 Advanced Grid

Use an *Advanced Grid* to place information or input fields clearly on the datasheet. The variables selected for the advanced grid are displayed as the grid columns. You can group the following variable types together in an advanced grid:

- Single inputline
- Date picker
- Single select
- Reference/Relation
- Relation property: Use a Relation to connect a specific variable from a different datasheet (an Asset Selector, for example).

  > Note: Note that you can only add a *Relation* if you have already created a *Relation* variable in the table.

- Organizational units

  > Note: You can only add the *Organizational units* variable type to the advanced grid once. If you add other variable types, the *Organizational units* entry is no longer displayed for selection.

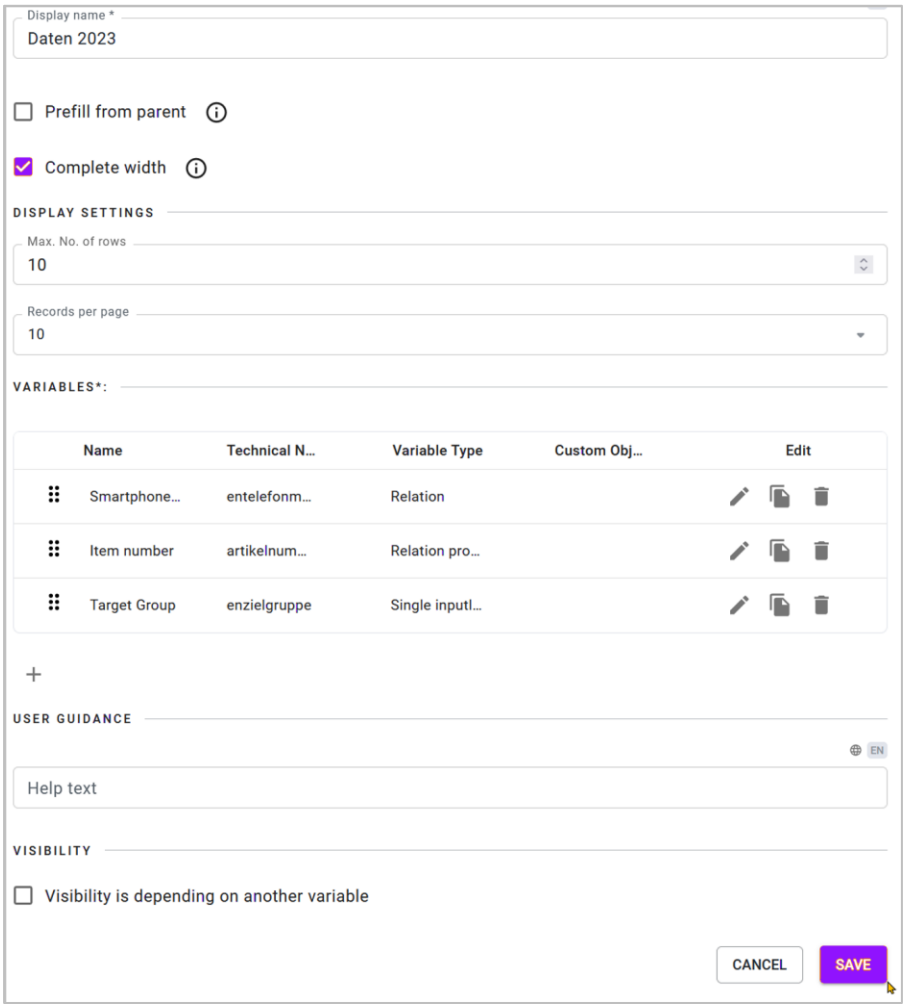
- Multiline input area
- Asset Selector
- Multiselect
- Combo box
- Action: An action button can be integrated into an extended table. Via the button, users call customer-specific functions. The button is not displayed in a column like other variables, but the user reaches the button when a row is in edit mode. Clicking the button invokes a URL stored in the variable.

Each variable type can be configured as a mandatory field, which means that the user must edit the field when populating the table. To accomplish this, set the checkbox at *Mandatory field* when creating.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. None of the fields are visible when you create the variable. See Display, technical, and unique name, page 24. Note: If you want the values of a parent table to be inherited by the table, the technical name of the parent and child tables must be identical.
Display name *	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Max. No. of rows	Define how many maximum rows the table can contain.
Table height	Enter the table height in pixels.
Complete width	Activate this checkbox to utilize the entire width of the datasheet. Note: If this checkbox is activated, the variable can only be placed on a one-column datasheet layout.
Records per page	Define with how many records per page the table is paginated: 10, 20, 50 or 100 records per page or without pagination. Note: Note that the performance for displaying large tables with pagination improves considerably.
Prefill from parent	Activate this checkbox if you want the table to be filled by a table in a higher-level job or data object. The user can edit the table in the child job or child data object and delete and add data records. The user can also refresh the data filled by the parent element. All changes are lost in this case. Note: For a successful inheritance, the parent and child tables must have an identical technical name.
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.
	You can add an additional variable to the grid.

Name	Description
Variables	Your selected variables are listed in the *Variables* area. You can: • Change the order of the variables. • Open a variable for editing. • Copy a variable. • Delete a variable and remove it from the table.



5.1.3 Asset Selector

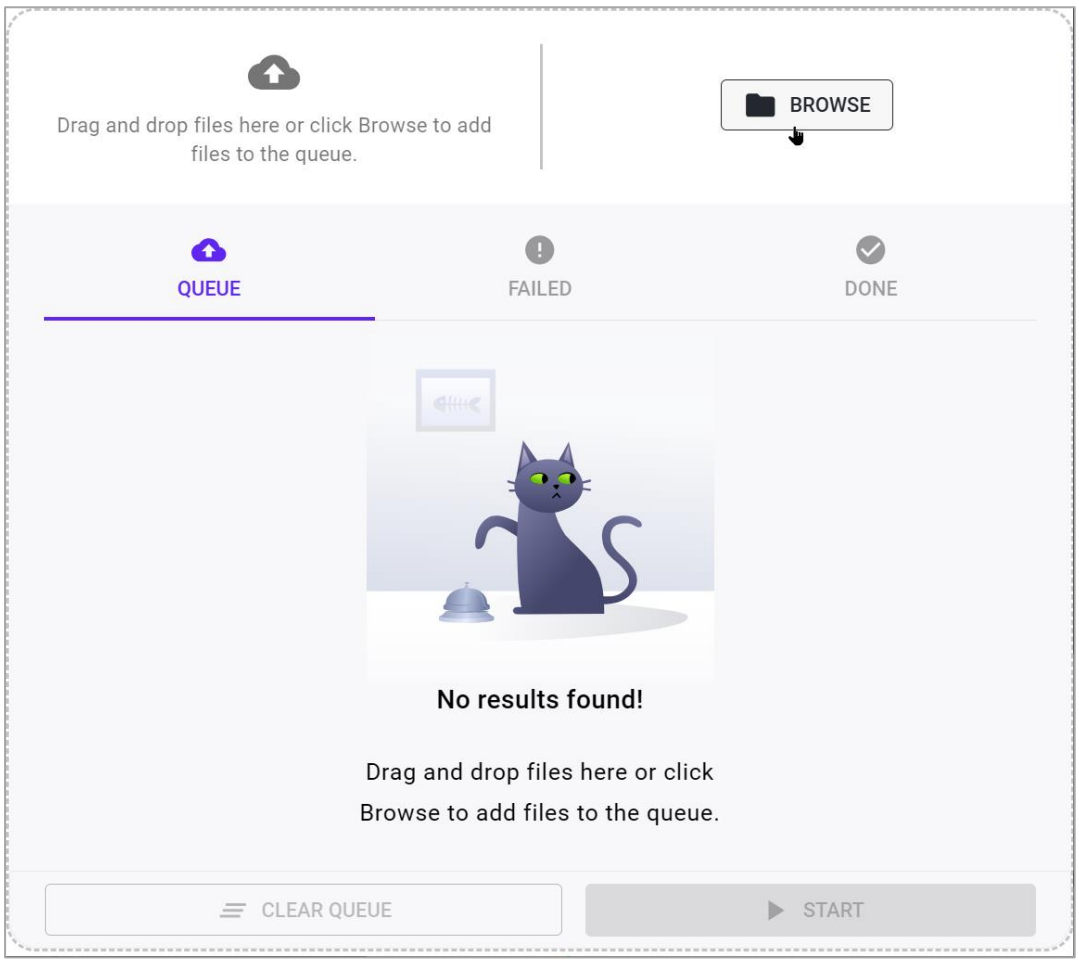
Use an *Asset Selector* to make the buttons *Upload new assets* and *Select from Media Pool* available on a datasheet. You can use the *Upload new assets* function to add both locally saved files and assets from a collection to the datasheet. You can also specify that assets are added to a datasheet automatically based on specific criteria.

> Note: the assets that can be selected and edited are determined by the permissions in the user role.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. None of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
*Display name**	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
Complete width	Checkbox, activate to *utilize the entire width of the datasheet.*
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.

Name	Description
Prefilling from custom structure	Select a custom structure and a created attribute. By choosing *Key*, you can configure the display of an asset based on the affiliate ID. You can use the option *Override with* to link the display of an asset to a single selection that is linked to the same custom structure: in the single selection, you choose one of the custom objects (for instance, the data object). The text field is linked to the attribute *Product image*. If a different data object is set in the single selection, the corresponding data object image is automatically displayed in the text field.
Select media automatically	Activate this checkbox to add assets with the following properties to a datasheet automatically: • *Show all assets with the same 'Item Number'* • *Show all media that include ID in 'free text field'*: Specify the free text field in which the job ID or data object ID must be entered. Note: If one of these options is activated, users cannot select assets manually.
Free text field	Note: This can only be used when the checkbox *Show all media that include ID in 'free text field'* is activated. You can specify the free text field in which the job ID or data object ID must be entered.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the variable only after a user has selected a specific value for a different selection field on the datasheet.
*Variable**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the selection field upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected in order to display the variable.



5.1.4 Combo Box

Use a *Combo box* to combine the features of a *Single select* and *Single inputline* with each other. In a combo box, users can:

- Select from values that are defined as a data source using a custom structure.

  > Note
  >
  > Stored values are proposed using Auto-Complete.

- Enter other values that are different from the stored values if required.

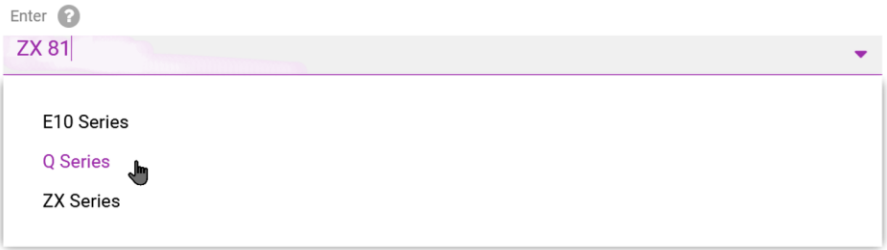
Prerequisites:

- You require a custom structure that you can select as the data source.

The following parameters are provided when you create or change the variables:

Name	Description
Unique Name *Technical Name*	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
*Display name **	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
*Data Source**	Select an existing custom structure as data source for the values of the selection field.
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.

Name	Description
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.
Default value	Enter the value to use as the default for the field. Note: Note that the default value is adopted only when you create an object (data object/job). Changing the default value does not have any effect on objects that have been created already.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the selection field only after a user has selected a specific value for a different selection field on the datasheet.
*Variable**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the selection field upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected to display the variable.



5.1.5 Comfort Grid

Users can also use the comfort grid to enter data in a clear tabular form. When users fill in the grid on the datasheet, numerous keyboard shortcuts in particular make editing easier. Please refer to the user manuals for details.

5.1.5.1 Data Types

You can use the following data for a comfort grid:

- One-line text field
- Numbers
- Single select: selection of a value from a structured list.
- Date
- Formula, for details see chapter

Each variable type, except formula. can be configured as a mandatory field, i.e., the user must edit the field when filling the table. To achieve this, set the checkbox at *Mandatory field* when creating.

Numerical and date values are displayed and entered localized according to the selected UI language:

5.1.5.2 Summary in Footer

You can set up a summary for each table variable in the footer of the table. To do this, enter a name and select one of the following functions:

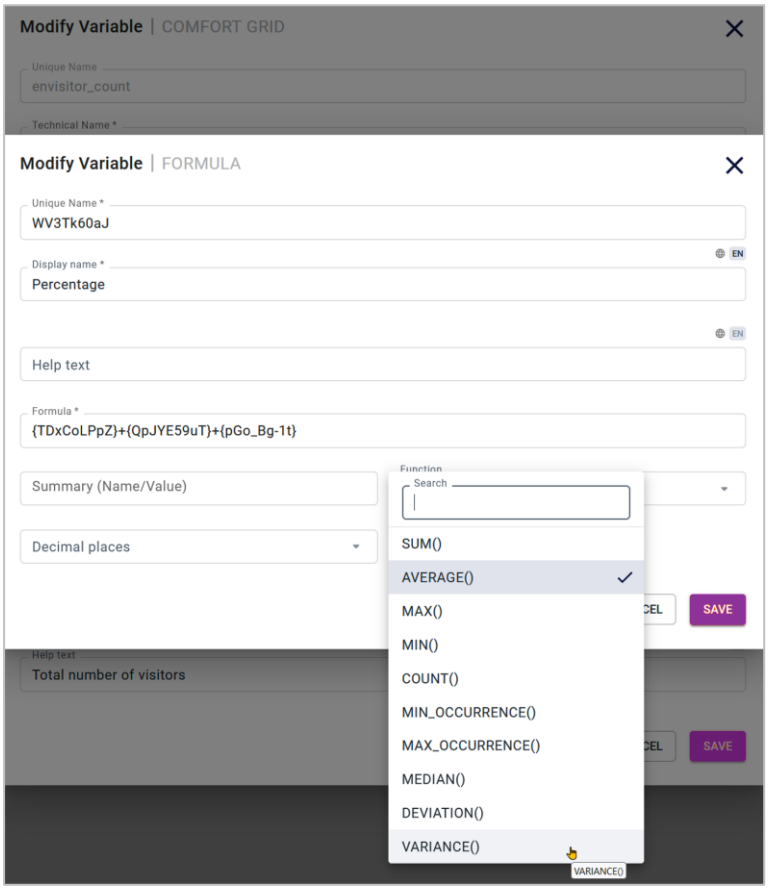
- SUM(): Sum of all numbers in the column
- AVERAGE(): arithmetic mean of all numbers in the column
- MAX(): highest value in the column
- MIN(): lowest value in the column
- COUNT(): number of values in the column
- MIN_OCCURRENCE():Value that occurs least in the column.

Example
The column is a variable of type Single-Selection with the values A, B, C and D. The column contains 5 times A, 1 time B, 0 times C and 3 times D. The result in the footer is C (0).
Comma-separated values are displayed if the result contains multiple values.

- MAX_OCCURRENCE(): Value that occurs most frequently in the column.

Example
The column is a variable of type Single-Selection with the values A, B, C and D. The column contains 5 times A, 1 time B, 0 times C and 3 times D. The result in the footer is A (5).
If the result is not unique, several values are displayed, separated by commas.

- MEDIAN(): Median of the values in the column
- DEVIATION(): Deviation, rounded to two decimal places
- VARIANCE(): Variance, rounded to two decimal places



You can combine the functions with table variable types as follows:

	Text	Date	Single Select	Number	Formula
SUM()				✓	✓
AVERAGE()				✓	✓
MAX()		✓		✓	✓
MIN()		✓		✓	✓
COUNT()	✓	✓	✓	✓	✓
MIN_OCCURRENCE()		✓	✓	✓	✓
MAX_OCCURRENCE()		✓	✓	✓	✓
MEDIAN()				✓	✓
DEVIATION()				✓	✓
VARIANCE()				✓	✓

5.1.5.3 Formula Variables

With a formula variable, you calculate values in a column according to an entered formula. The following operators and operands are available:

- +, -, /, * and (and)
- Numbers
- Other variables of the Number type within the table. You address the variables as follows: {Name of the variable}. Then the value of the variable is copied from the same row into the formula. Note Example 1 below.
- The following summaries for other columns. Note Example 2 below:
    - SUM(): sum of all numbers in the column
    - AVERAGE(): arithmetic mean of all numbers in the column
    - MAX(): highest value in the column
    - MIN(): lowest value in the column
    - COUNT(): number of values in the column
    - MEDIAN(): Median of the values in the column
    - DEVIATION(): Deviation, rounded to two decimal places
    - VARIANCE(): Variance, rounded to two decimal places
- Mathematical operations and constants according to the following Library; note Example 3 below.

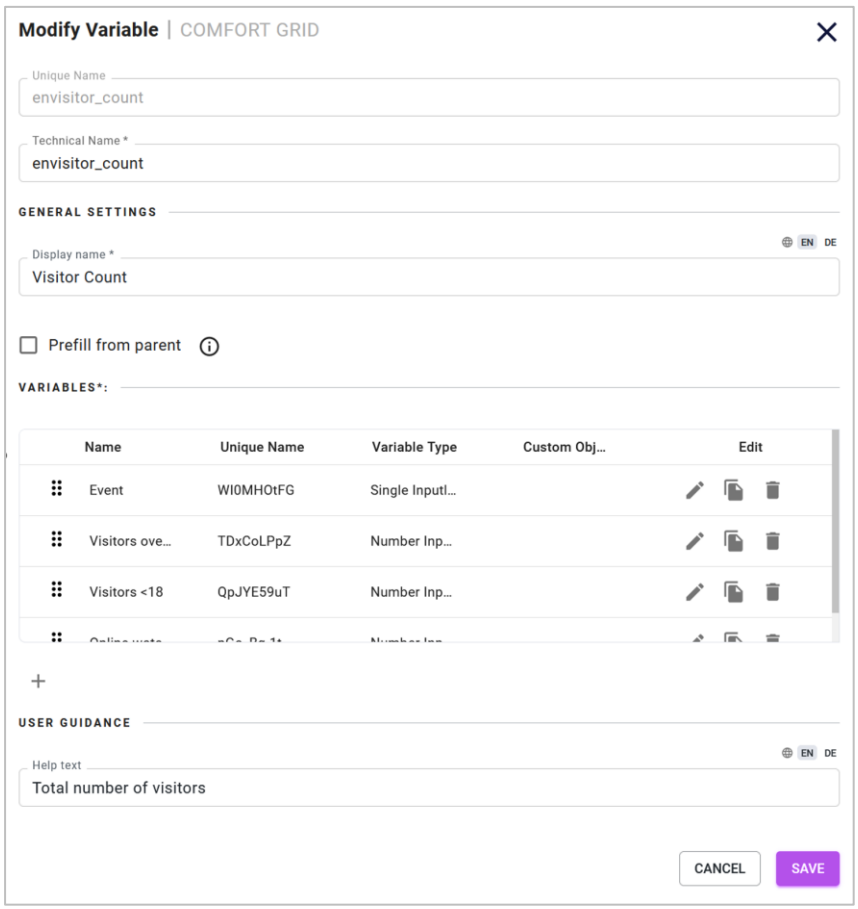
Example 1

A table is to be used to enter visitor data for events, categorized by visitors over 18 years of age, visitors under 18 years of age, and visitors who participated in the event online. For this, you need the average in the respective category over all events. Additionally, you want to calculate the following values:

- Total number of visitors
- Percentage of online visitors

For these values, you also want to know the average across all events.

Implementation: Create a variable of type *Comfort Grid* on the job type datasheet. Add the following column variables to the table:

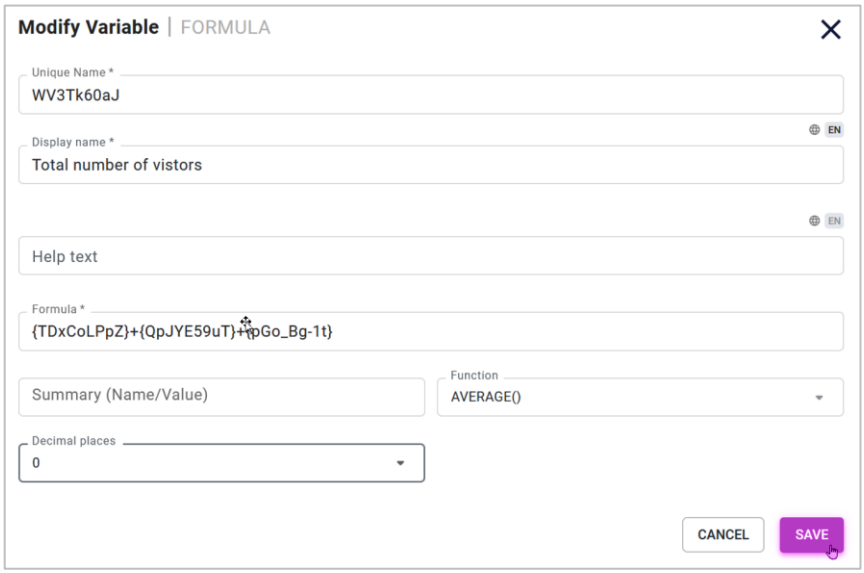


- Event: simple text field, summary function: COUNT() to get the total number of events.
- Visitors over 18: simple number field to enter the number of visitors over 18 years old, summary function: AVERAGE()
- Visitors U18: simple number field to enter the number of visitors under 18 years old, summary function: AVERAGE()

- Online Visitors: simple number field to enter the number of visitors who participated online; summary function: AVERAGE()

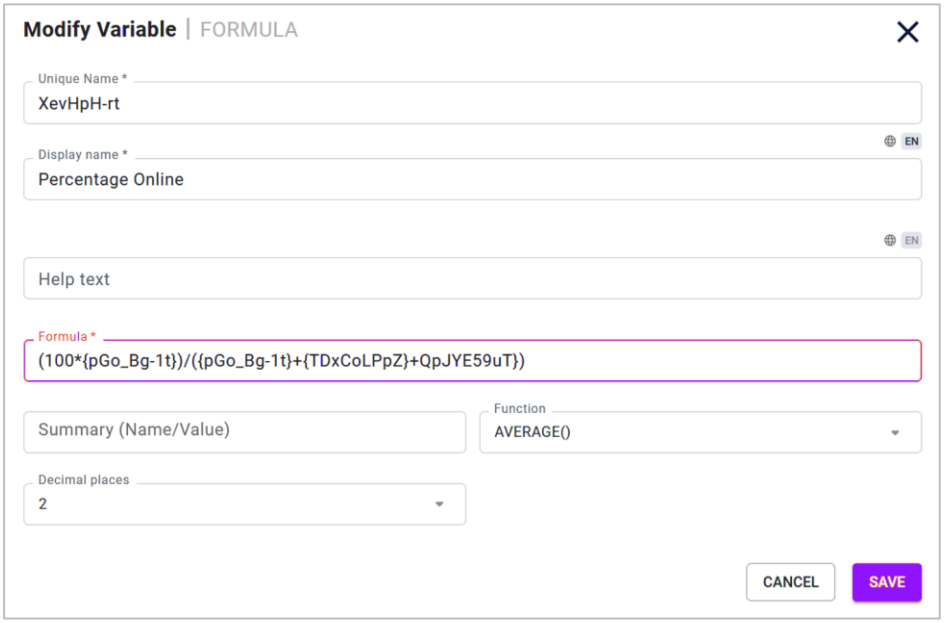
These fields are edited for each event by the user on the datasheet. To calculate the other values, create two formula variables in the comfort grid:

- Total number of visitors: The formula is
  {Visitors over 18}+{Visitors U18}+{Online Visitors}.
  However, you must store formulas with unique names (e.g.,: WI0MHOtFG) for the variables. To achieve this, first create a human-readable formula and then replace the names to avoid unintentional errors. There is no syntax check. If a formula is not correct, you will only recognize this when created jobs cannot be opened.
  The formula sums up the entered values in the number fields for each event in the respective line. Note that the unique names in the curly brackets must correspond to the names of the number field variables described above.

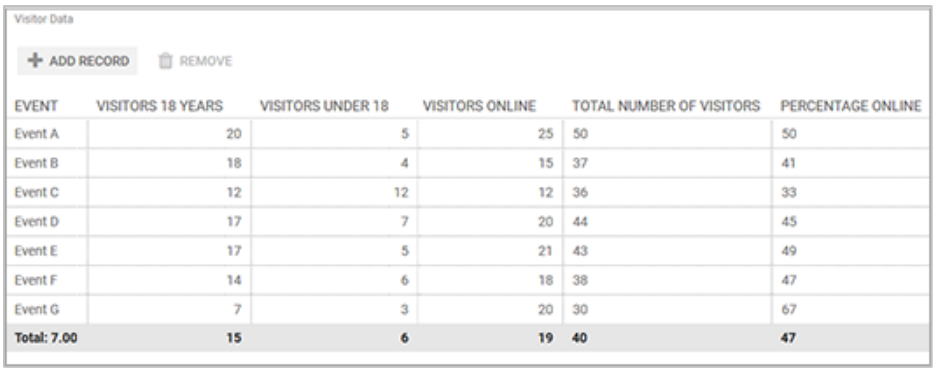


- Share Online: The formula is
  (100*{Visitors Online})/({Visitors Online}+{Visitors O18}+{Visitors U18}).

Again, replace all names in curly brackets with the unique names.



The following screenshot displays how a correspondingly filled table is displayed on the job datasheet.



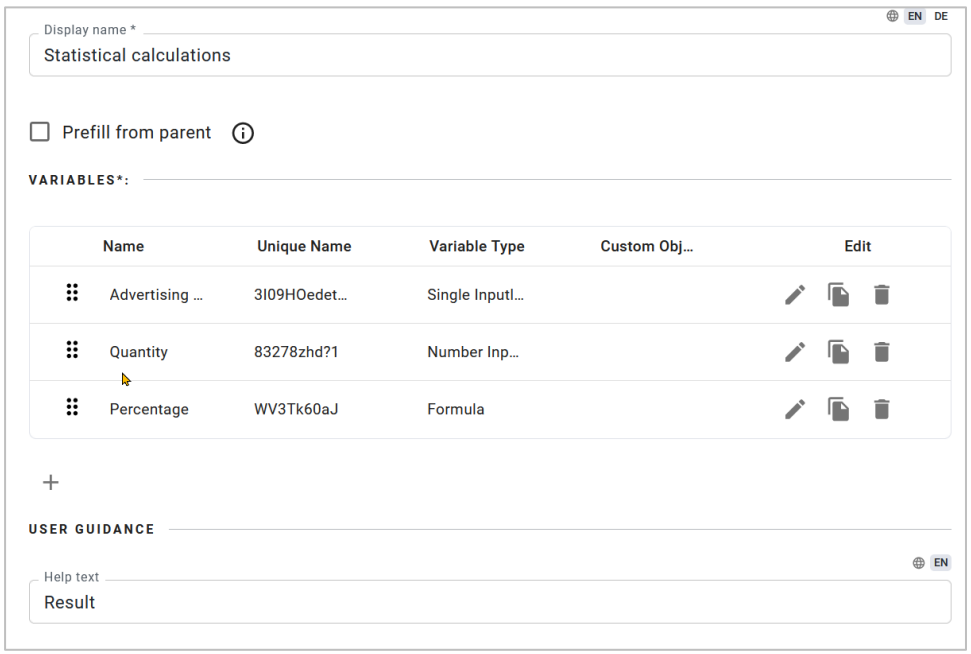
EVENT	VISITORS 18 YEARS	VISITORS UNDER 18	VISITORS ONLINE	TOTAL NUMBER OF VISITORS	PERCENTAGE ONLINE
Event A	20	5	25	50	50
Event B	18	4	15	37	41
Event C	12	12	12	36	33
Event D	17	7	20	44	45
Event E	17	5	21	43	49
Event F	14	6	18	38	47
Event G	7	3	20	30	67
Total: 7.00	**15**	**6**	**19**	**40**	**47**

In this case, the user only fills in the first four columns. The last two columns are calculated automatically.

Example 2

You want to compile and evaluate the number of distributed advertising media in a table.

Implementation: Create a variable of type *Comfort Grid* on the job type datasheet. Add the following column variables to the table:

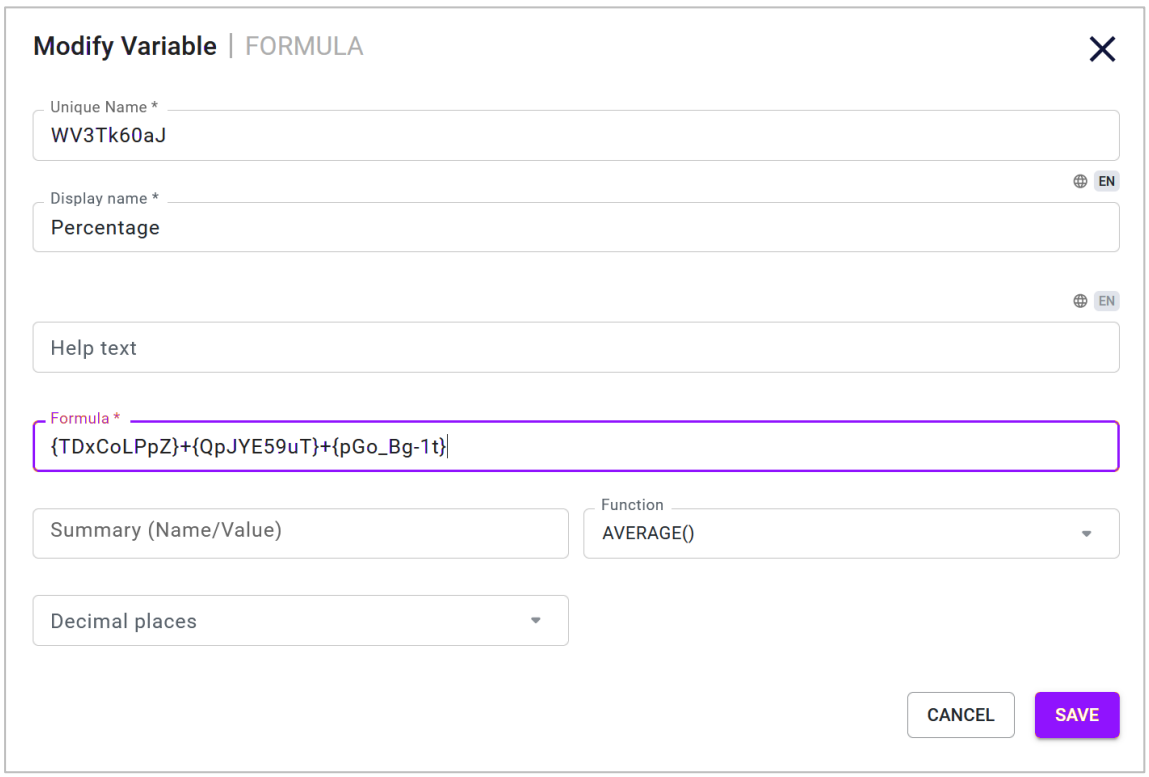


- Advertising media: Simple text field, without summary function
- Number: Simple number field to enter the number of ad media, summary function: SUM()

These fields are edited for each ad media by the user on the datasheet. To calculate the percentage, create a formula variable in the comfort grid:

- Ratio: The human-readable formula is {Number}*100/SUM({Number})

  Replace *Number* in curly brackets with the unique name. The formula takes the entered number of the ad media in the respective row and calculates the share of the sum of all values entered in the Number column. The result is given with one decimal place.



Example 3

With the following Library (external link) you can insert mathematical functions and constants in the formulas. You can insert the functions and constants into the formulas as follows:

- Example Pi: You want to multiply a value from the Diameter column by Pi. The entry in the formula line is: Math.PI{Diameter}
- Example Powers: You wish to take the square root of the sum of two column values: Math.pow({column A}+{column B}, 0.5)

5.1.5.4 Grid Parameters

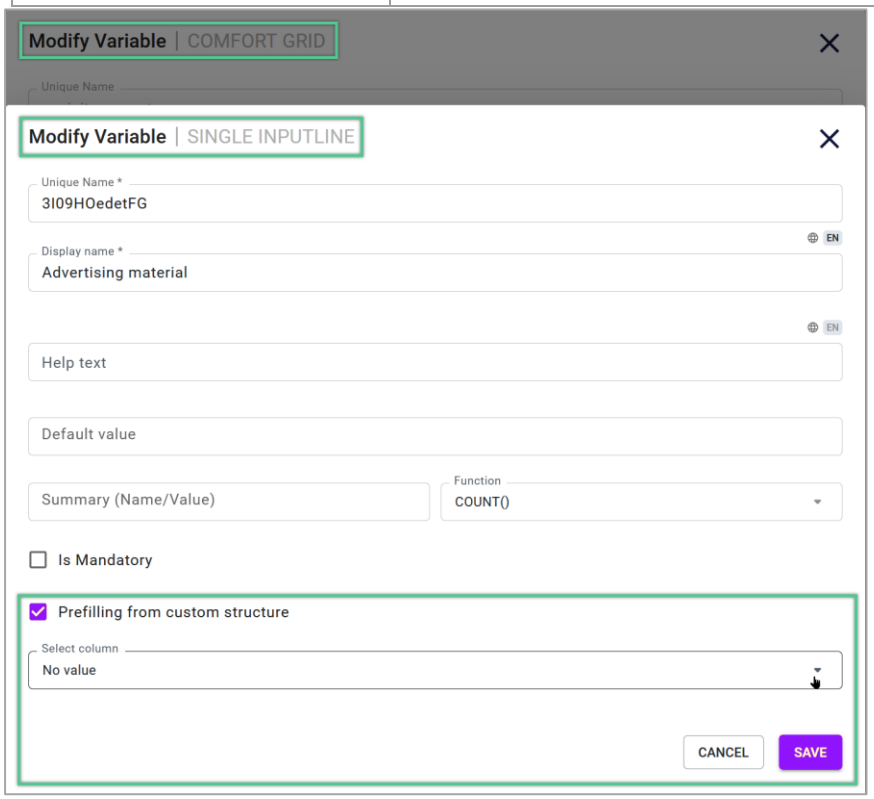
The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: this is visible only if the variable is changed. Neither of the fields are visible when you create the variable. For details, see chapter 2.2.
Display Name	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. You can create required language versions. For details, see chapter 2.2.
Prefill from parent	Activate this checkbox if you want the table to be filled by a table in a higher-level job or data object. The user can edit the table in the child job or child data object and delete and add data records. The user can also refresh the data filled by the parent element. All changes are lost in this case. Note: For a successful inheritance, the parent and child tables must have an identical technical name.
Help text	Define the help text that can be displayed to users. You can create required language versions.
+	You can add an additional variable to the grid. Simple text fields, number fields and date fields have the option for *Pre-filling from custom structure* within the comfort grid. See above in the table under 5.1.14.
Variables	Your selected variables are listed in the *Variables* area. You can: • Change the order of the variables. • Open a variable for editing. • Copy a variable. • Delete a variable and remove it from the table. For a description of the parameters for the various table variables, see the following chapter 5.1.5.5.
Prefilling from custom structure	Activate the checkbox to be able to select already existing values of the type *Single select* here. After you select a column, you can select an attribute in the field below it. There, only attributes of type date, text, floating-point number and integer can be selected.

5.1.5.5 Grid Variables

Single Inputline Variable

Name	Description
Unique name *Technical Name*	Note: this is visible only if the variable is changed. Neither of the fields are visible when you create the variable. For details, see chapter 2.2.
Display Name	Define the name with which the variable is displayed in the table. You can create required language versions.
Help text	Enter the help text that can be displayed for a user. You can create required language versions.
Default value	Enter a default value.
Summary	Enter a designation in the left field. Select a function for the summary in the right field, see chapter 5.1.5.2.
Mandatory	Activate the checkbox if the user has to edit the column.
Prefilling from custom structure	Activate the checkbox to be able to select already existing values of the type “Single selection” here. After you select a column, you can select an attribute in the field below it. There only attributes of type date, text, floating-point number and integer can be selected.



Simple text fields, simple number fields and date selections can now be prefilled in a comfort grid.

Date Picker Variable

Name	Description
Unique name *Technical Name*	Note: this is visible only if the variable is changed. Neither of the fields are visible when you create the variable. For details, see chapter 2.2.
Display Name	Define the name with which the variable is displayed in the table. You can create required language versions.
Help text	Define the help text that can be displayed to users. You can create required language versions.
Summary	Enter a designation in the left field. Select a function for the summary in the right field, see chapter 5.1.5.2.
Mandatory	Activate the checkbox if the user has to edit the column.
Prefilling from custom structure	Activate the checkbox to be able to select already existing values of the type “Single selection” here. Select an existing custom structure and a created attribute whose values you want to use to prefill the date picker.

Single Select Variable

Name	Description
Unique name *Technical Name*	Note: this is visible only if the variable is changed. Neither of the fields are visible when you create the variable. For details, see chapter 2.2.
Display Name	Define the name with which the variable is displayed in the table. You can create required language versions.
Source	Choose a custom structure as source for the dropdown list.
Dependent on parent selection	Note: Only visible if a single selection has already been created in the table, which is linked to the custom structure superior to the data source. Activate the checkbox if the display of the selections in this field depends on the setting in another selection field, for example, the display of street names on the selection of a town in another selection field.
Parent selection	Note: This is visible only if the *Dependent on parent selection* checkbox is activated. Select the parent field containing the setting that the displayed values of the selection depend on.

Name	Description
Help text	Enter the help text that can be displayed for a user. You can create required language versions.
Default value	Select a default value.
Summary	Enter a designation in the left field. Select a function for the summary in the right field, see chapter 5.1.5.2.
Mandatory	Activate the checkbox if the user has to edit the column.

Single Number Field Variable

Name	Description
Unique name *Technical Name*	Note: this is visible only if the variable is changed. Neither of the fields are visible when you create the variable. For details, see chapter 2.2.
Display Name	Define the name with which the variable is displayed in the table. You can create required language versions.
Help text	Define the help text that can be displayed to users. You can create required language versions.
Default value	Enter a default value.
Summary	Enter a designation in the left field. Select a function for the summary in the right field, see chapter 5.1.5.2.
Decimal places	Set the number of decimal places with which the data in the column will be displayed. Possible are 0 to 10 decimal places.
Mandatory	Activate the checkbox if the user has to edit the column.
Prefilling from custom structure	Activate the checkbox to be able to select already existing values of the type “Single selection” here. Select an existing modifiable structure and a created attribute with whose values the number field will be prepopulated.

Formula Variable

Name	Description
Unique name *Technical Name*	Note: this is visible only if the variable is changed. Neither of the fields are visible when you create the variable. For details, see chapter 2.2..
Display Name	Define the name with which the variable is displayed in the table. You can create required language versions.
Help text	Define the help text that can be displayed to users. You can create required language versions.
Formula	Enter the formula.
Summary	Enter a designation in the left field. Select a function for the summary in the right field, see chapter 5.1.5.2.
Decimal places	Set the number of decimal places used to display the data in the column. Possible are 0 to 10 decimal places.

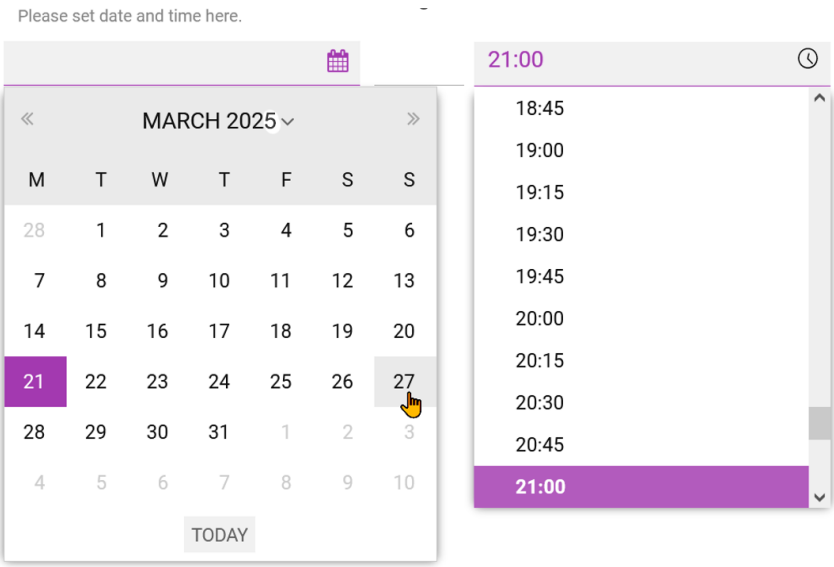
5.1.6 Date Picker With and Without Time

Use the *Date picker* variable to create a date field with a date picker on the datasheet. The variable *Date picker with time* also provides an additional field for the time.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
Display name *	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
For all types	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
Validity	Select the checkbox *Allow only valid dates* to define conditions for the date selection validity.
Number of offset days	Note: This is visible only if the *Validity* checkbox is activated. Specify the minimum number of days into the future the date must be in relation to the *Reference* date.
Reference	Note: This is visible only if the *Validity* checkbox is activated. Define the date to which the Number of offset days refers. Choose: • *Creation date* • *Current date* • *Transfer date from initial creation to any other workflow step*
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.

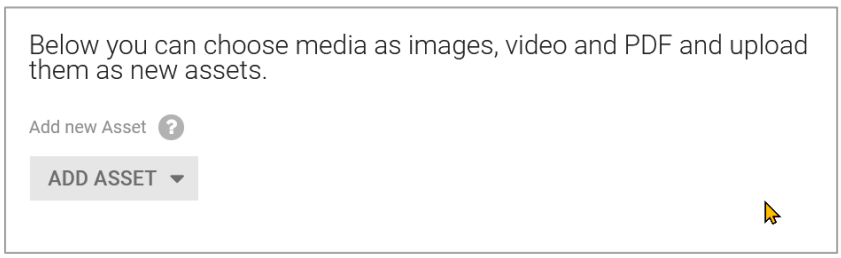
Name	Description
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.
Prefilling from custom structure	Note: This option can only be used for the *Datepicker* variable. Select an existing custom structure and a created attribute whose values you want to use to prefill the date picker.



5.1.7 Description Text

Use a *Description text* to enter additional information on a datasheet, for example.

Name	Description
Unique name, technical name	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
Text	This displays the title used to display the headline text on the datasheet. You can create any language versions that are required. The text is used as the display name in the system; see Display, technical, and unique name, page 24.
Visibility	Activate the checkbox *Visibility is depending on another variable* to display the field only after a user has selected a specific value for a different selection field on the datasheet.
Variable	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the dropdown list upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected in order to display the variable.



5.1.8 Document Selector

Use a *Document Selector* to make the *Select from Brand Template Builder* button available on a datasheet. Users can use this button to open a search in *the Brand Template Builder* module and add a document (finalized or still in progress) to the datasheet.

A document that is still in progress can be loaded to the document wizard for further processing directly from the datasheet. Users can open the detailed view of a finalized document and call additional functions (to adjust it to suit their own purposes, for example).

> Note: Which documents in *Brand Template Builder* module can be selected and edited is determined by the permissions in the user role.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
*Display name**	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.

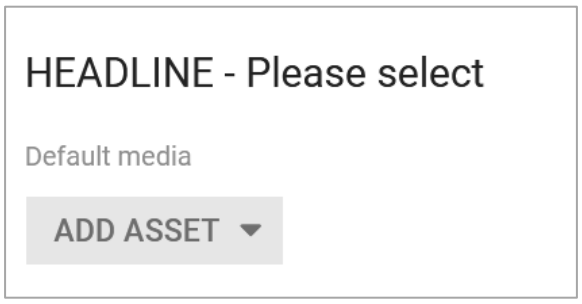
Name	Description
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the selection field only after a user has selected a specific value for a different selection field on the datasheet.
*Variable**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the selection field upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected to display the variable.

5.1.9 Headline Text

Use a *Headline Text* to name the sections of a datasheet and structure the placed variables, for example.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. None of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
Text	This displays the title used to display the headline text on the datasheet. You can create any language versions that are required. The text is used as the display name in the system; see Display, technical, and unique name, page 24.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the field only after a user has selected a specific value for a different selection field on the datasheet.
Variable	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the dropdown list upon whose value you want to make the visibility dependent.
Value	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected to display the variable.

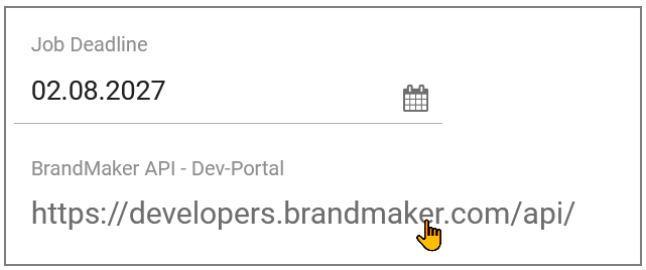


5.1.10 Link

You use a *Link* to create a reference to another page in the Marketing Efficiency Cloud (e.g., a Marketing Shop page) or an external website.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
Display name *	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type	You can create two different types of links: • *Relative*: This creates a link to an existing page in the Marketing Efficiency Cloud (for example, a Marketing Shop page. Do not enter the entire address as the link; instead, enter only the part that is relevant to the link. The "front" part of the link (the URL of your system) is generated and added automatically. • *Absolute*: This creates a link to an external web page.
Help text	Define the help text that can be displayed to users.
URL	Enter the address of the (web) page for which you want to create a link. Note: You must enter the full address for an external web page (including "http://" or "https://").
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the selection field only after a user has selected a specific value for a different selection field on the datasheet.
*Variable**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the selection field upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected to display the variable.



5.1.11 Multiple Values Input Lines

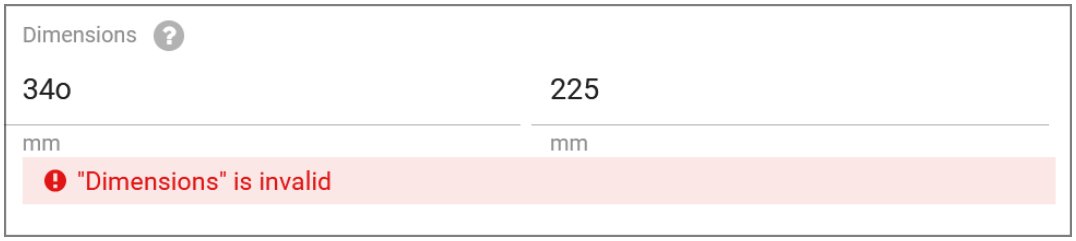
Use the variable *Multiple values input lines* to place a field for two related values (height and width, for example) next to each other on a datasheet.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. None of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
Display name *	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.
Percentage	Distribute the width of both input fields in percentage to the total width of the datasheet.
Max. Characters	Define the maximum number of characters that can be entered in the field.
Reg. Exp. Validator	Enter a regular expression in the field to check the entry for validity. For example, you can then ensure that bank codes or e-mail addresses have been entered in a specific format.
Input Size	Define the width of the variables. If so many characters are entered that cannot be displayed in the configured width, the field is provided with a scroll bar.

Name	Description
Suffix	You can enter a suffix (`Millimeter` or `%`, for example) that is attached to the field.
Help text	Define the help text that can be displayed to users.
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.
Default value	Enter the value to use as the default for the field. Note: Note that the default value is adopted only when you create an object (data object/job). Changing the default value does not have any effect on objects that have already been created.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the field only after a user has selected a specific value for a selection field on the datasheet.
*Variable**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the selection field upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected to display the variable.

In this example, the input is invalid because the regex default, which only allows three-digit numbers, is not met and the lowercase letter “o” was entered instead of zero.



5.1.11.1 Examples of Using Regular Expressions

When creating or modifying the expressions, you can use an online tool such as https://regexr.com/ for testing. The following table provides you with some proven examples for a successful start. For further support, please visit the above website.

Name	Description
E-mail address	`^[\w.+-]{2,64}\@[\w.-]{2,249}\.[a-z]{2,6}$`
SWIFT BIC Code	`/^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?$/`
IBAN without spaces	`[a-zA-Z]{2}[0-9]{2}[a-zA-Z0-9]{4}[0-9]{7}([a-zA-Z0-9]?){0,16}`
Three-digit number	`\d{3}`
German zip code	`[0-9]{5}`
IPv4 address	`^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`

5.1.12 Numbers

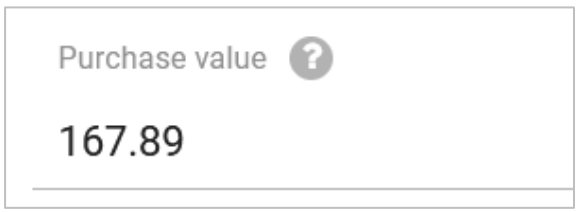
Use the *Numbers* variable to make an input field for numbers available to users on the datasheet.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
Display name *	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
Help text	Define the help text that can be displayed to users.
Use Unit	Select this checkbox to add a unit to the field.

Name	Description
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.
Unit of measure	Note: This is visible only if the *Use Unit* checkbox is activated. You can select whether a unit for *Length* or *Weight* is displayed.
Default unit	Note: This is visible only if the *Use Unit* checkbox is activated. You can define a default entry for the unit of measure, such as centimeter (cm) or kilogram (kg).
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the variable only after a user has selected a specific value for a different selection field on the datasheet.
Variable	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the dropdown list upon whose value you want to make the visibility dependent.
Value	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected to display the variable.

If the number field is enabled with the option *Use Unit*, it can also run conversions between units for you. For comparison, you can see a simple number field below.



5.1.13 Relation

Use a *Relation* to link datasheets to each other. For example, users select a job from a selection list. Users can then open the relevant datasheet directly from the displayed link.

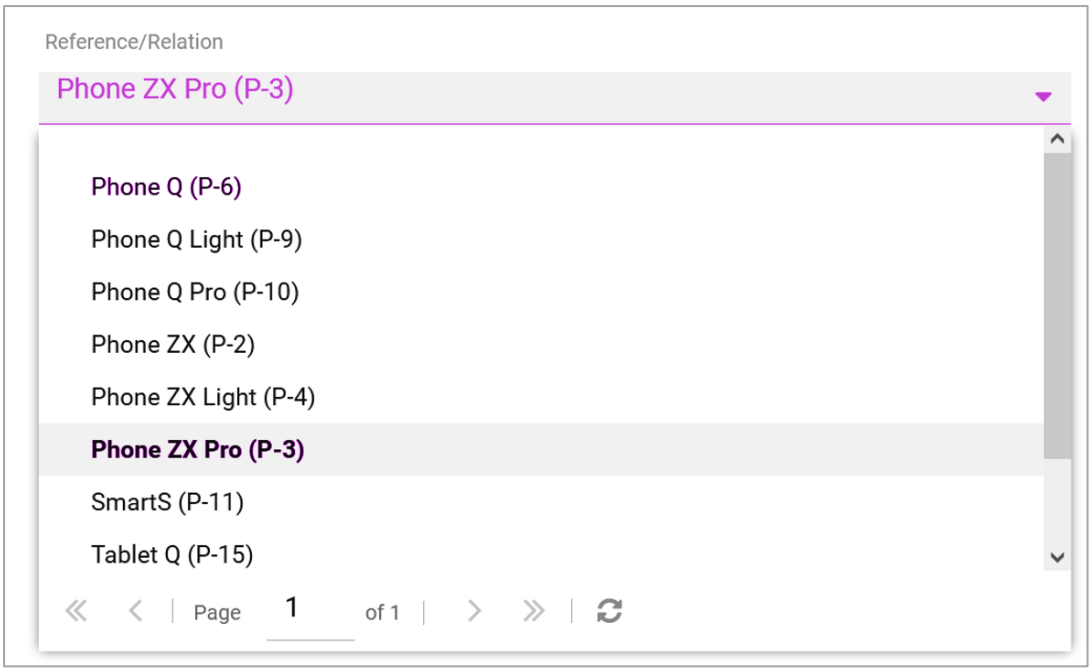
Prerequisite

- You require at least one type that you can reference.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. None of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
*Display name **	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
*Module**	This selects the module containing the types to which you want to make the reference.
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.
Type	All types available based on the module you selected are displayed. You can select the types you want to reference.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the variable only after a user has selected a specific value for a different selection field on the datasheet.

Name	Description
*Variable**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the dropdown list upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected to display the variable.



5.1.14 Single Inputline vs. Multiline Input Area

Use *single inputline* and/or *multiline input area* text fields to allow users to enter text on the datasheet.

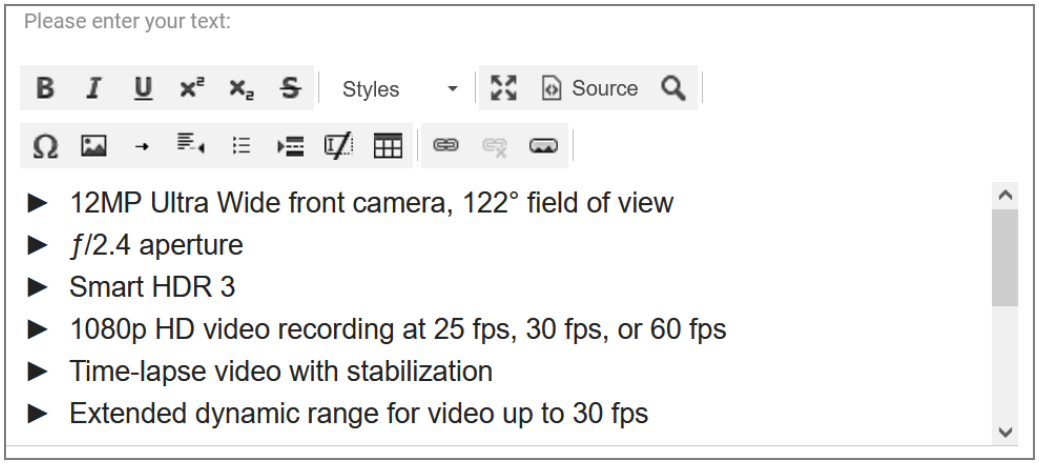
> Note
>
> You can enter a maximum of approximately 500,000 characters in a multiline input area.

The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
Display name *	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
Max. Characters	Define the maximum number of characters that can be entered in the field.
Columns Visible	Note: Visible only for multiline input areas. You can define the width of the input area by specifying the number of columns.
Lines Visible	Note: Visible only for multiline input areas. You can define the height of the input area by specifying the number of lines.
Complete width	Note: This is visible only for multiline input areas. Activate this checkbox to utilize the entire width of the datasheet for the input area in a one-column layout. If you activate this checkbox, leave the *Columns Visible* field empty.
Allow formatting	Note: Visible only for multiline input areas. This checkbox can only be activated while you create the variables. This displays a rich-text editor that allows users to format the text (for example, bold, italics, underline, and so on).

Name	Description
Editor configuration	Note: This is visible only for multiline input areas and if the *Allow formatting* checkbox is activated. You can select an editor configuration from the selection list. This allows you to define which formatting the user can use in the text field. For more information, see the configuration manual.
Reg. Exp. Validator	Note: Visible only for single input lines. Enter a regular expression in the field to check the entry for validity. For example, you can ensure that entries for BIC codes or e-mail addresses are provided in a specific format.
Input Size	Note: Visible only for single input lines. Defines the maximum number of characters displayed. If more characters are entered, the field gets a scroll bar.
Suffix	Note: Visible only for single input lines. You can enter a suffix (`Millimeter` or `%`, for example) that is attached to the field.
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.
Default value	Enter the value to use as the default for the field. Note: Note that the default value is adopted only when you create an object (data object/job). Changing the default value does not have any effect on objects that have been created already.

Name	Description
Prefilling from custom structure	Select a custom structure and a defined attribute of this structure. By choosing *Key*, you can configure the pre-assignment of a value based on the branch number of the current user or on the name of the user. In the process, the unique names of the custom objects must match the branch number or the username. You can use the option *Override with* to link the display in the text field to a single selection that is connected to the same custom structure: In the single selection, you select one of the custom objects (for instance, the branch office). The text field is linked to the attribute *Address*. If a different affiliate is set in the single selection, the corresponding address is automatically displayed in the text field. If you activate the checkbox *Activate a fixed connection to the selected "custom object"*, the text field content is updated automatically if the custom structure is changed at a central location. Note that to do so, a value must be selected in the *Override with* selection list.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the field only after a user has selected a specific value for a selection field on the datasheet.
Variable	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the selection field upon whose value you want to make the visibility dependent.
Value	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected to display the variable.



5.1.15 Single Select and Multiselect

Use a *Single select or Multiselect* to allow users to select one or more predefined values.

Prerequisite

- You require a custom structure that you can select as the data source.

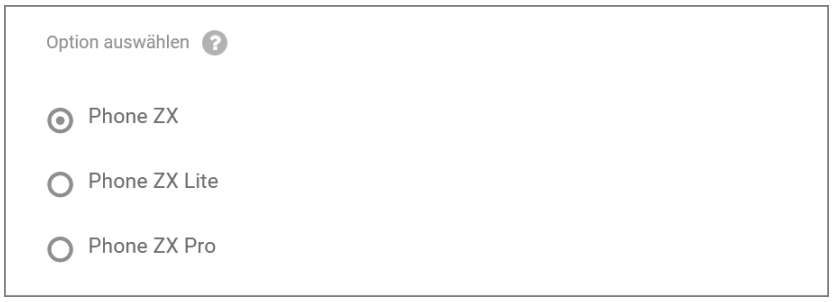
The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
*Display name **	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
Selection Type	You can define the layout for the selection field. *Single Select:* • *Dropdown*: This creates a dropdown list. • *OptionBoxArea*: This creates an option box. • *OptionBoxAreaWithImages*: This creates an option box. The existing preview images for the custom structure can be displayed. *Multiselect:* • *Selection Box*: This creates a selection list. • *OptionBoxArea*: This creates an option box. • *OptionBoxAreaWithImages*: This creates an option box. The existing preview images for the custom structure can be displayed.
Data source	Select an existing custom structure as data source for the values of the selection field.

Name	Description
Dependent on parent selection	Note: Only visible if there is a single selection of the *Dropdown* type or a multiple selection of the *Selection Box* type and if a parent selection has already been created on the datasheet. Activate the checkbox if the display of the selections in this field depends on the setting in another selection field, for example the display of street names on the selection of a town in another selection field.
Parent selection	Note: This is visible only if the *Dependent on parent selection* checkbox is activated. Select the parent field containing the setting that the displayed values of the selection depend on.
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.
Default value	Enter the value to use as the default for the field. Note: Note that the default value is adopted only when you create an object (data object/job). Changing the default value does not have any effect on objects that have been created already.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the selection field only after a user has selected a specific value for a different selection field on the datasheet.
*Variable**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the selection field upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected in order to display the variable.

Selection with *Multiselect*, display type: *SelectionBox*

Selection with *Singleselect*, display type: *OptionBoxArea*



5.1.16 Template

Use a *Template* variable to add a template from *Brand Template Builder* module to the datasheet. You can link the text boxes from the template to the variables on the datasheet to fill the corresponding text boxes automatically when editing the documents. Choose *Edit document* to open the document. If the values of the variables linked to the text boxes have been changed, a separate dialog box is displayed in which you can decide which content elements from the document are to be updated.

> Note
>
> A *BTB template* can be placed only on a one-column datasheet layout. To edit a document, you must have the corresponding rights.

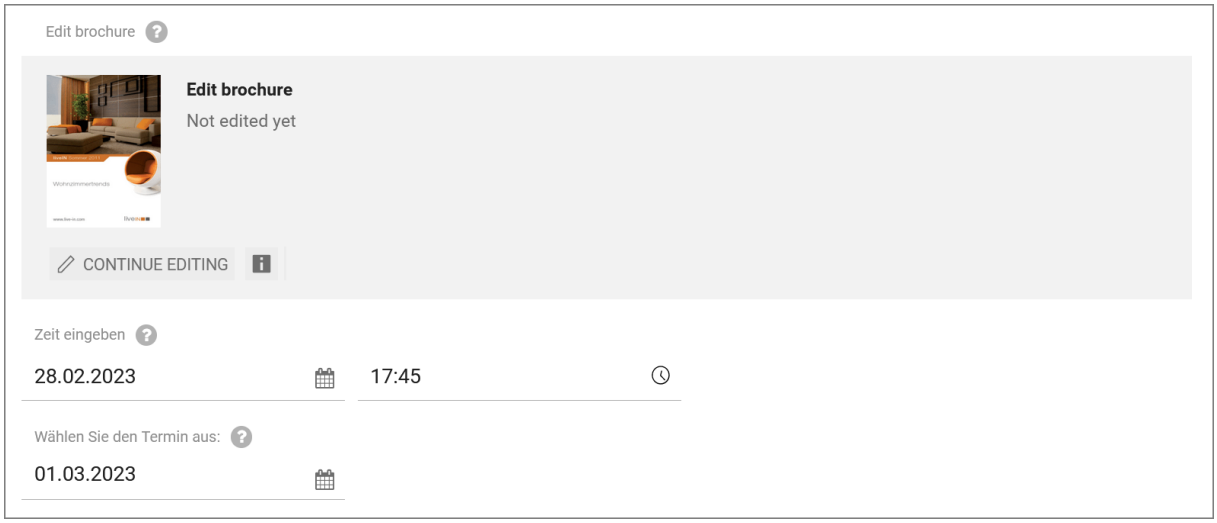
The following parameters are provided when you create or change the variables:

Name	Description
Unique name *Technical Name*	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. See Display, technical, and unique name, page 24.
*Display name**	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. See Display, technical, and unique name, page 24.
Type spanning	Select this checkbox to display variables with an identical technical name and identical variable type in one column in the filter view. See Grouping variables, page 64.
*Template ID**	Enter the ID number (excluding T-) for the template that you want to use.
Help text	Define the help text that can be displayed to users.
Shared value	Note: This is visible only if the type is a data object type and localization is enabled (see Localization, page 34). Note: Note that the variable can be selected as language-neutral only while you create it. The *Shared value* field cannot be changed at a later stage! Activate the checkbox if you want the variable value to receive the same value in each localized data object version.

Name	Description
Inherit from parent	Note: This is visible only if the type permits inheritance (*Inheritance* dropdown list = `Multiple parents`). Activate this checkbox if you want the data record to adopt the values of a parent job or data object as a sub-job or sub data object. Note: The variables must have identical display names.
Preview	As soon as you have entered a valid ID of a template, a preview image of the template is displayed.
Assign variables to boxes	You can assign a datasheet variable to the text boxes of the template that is in use.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the variable only after a user has selected a specific value for a different selection field on the datasheet.
*Variable**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can choose the dropdown list upon whose value you want to make the visibility dependent.
*Value**	Note: This is visible only if the checkbox *Visibility is depending on another variable* is activated. You can specify which value must be selected in order to display the variable.

Note

As soon as a new Web-to-Publish document is created, a preview of the created document is displayed on the datasheet.



5.1.17 User

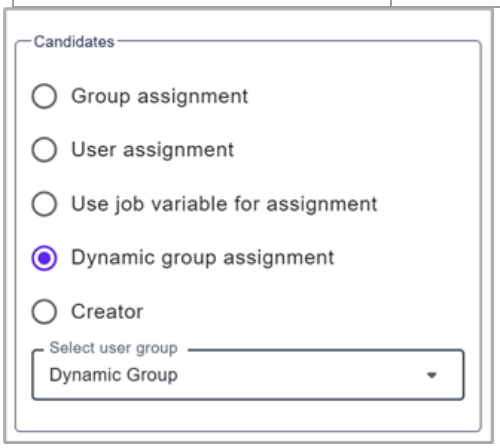
You can use the *User* variable to ensure that the user can select another system user on the datasheet. If necessary, you can limit the selection to users in a user group.

Name	Description
Unique name, technical name	Note: This is visible only if the variable is changed. Neither of the fields are visible when you create the variable. For details, see chapter 2.2.
Display Name	Define the name with which the variable is displayed on the datasheet. You can create any language versions that are required. For details, see chapter 2.2.
User group	Specify a user group. Once defined, the processor can only select users from this group on the datasheet. If you do not define a group, the user can choose from all the available users.
Default value	Enter the value to use as the default for the field. Note: Note that the default value is adopted only when you create an object (data object or job). Changing the default value does not have any effect on objects that have been created already.
Help text	Define the help text that can be displayed to users.
Visibility	Note: This is visible only if a single-select or multi-select is already created for the type. Activate the checkbox *Visibility is depending on another variable* to display the selection field only after a user has selected a specific value for a different selection field on the datasheet.

5.1.18 User Group

Use the *User group* field if you want to use the *Dynamic group assignment* function for user tasks in a BPMN workflow. If you do not preset the field with a default value, it must be marked as a required input field so that the workflow can be executed without errors later.

Name	Description
Unique name, technical name	Note: Only visible if variable is changed. Both fields are not visible when the variable is created. s. Displayed, technical and unique name, see chapter 2.2.
Displayed name	You define the name with which the variable is displayed on the data sheet. You can create the required language versions. s. Displayed, technical and unique name, see chapter 2.2.
Default value	You define a user group as the default setting. If the field is not filled in, the user must make a selection later.
Help text	Define the help text that can be displayed to users. You can create the required language versions.



5.2 System Variables

When you create a new type, variables are created and placed on the datasheet automatically. Absolutely vital system variables include:

- Job/data object name
- Job/data object type
- Workflow

Other system variables can be removed from the datasheet but cannot be deleted.

> Note
>
> The technical names of the system variables are used in the list. There may be a different name in the *Display name* field.

Technical Name	Variable type/Description
WorkflowTiming	Workflow timings: This variable is used to display the start date, duration, and due date of the current workflow step.
ArticleNumber	Single inputline: This variable can be used to display an existing item number (for an asset, for example).
Owner	Bean property: This variable is used to display the assignee.
Description	Multiline input area: This variable provides an input field. By default, the maximum number of characters is limited to 1000 and the height is defined as three lines.
CreateDate	Datepicker: This variable is used to display the creation date.
Creator	Bean property: This variable is used to display the creator of the job or data object.
Deadline	Datepicker: This variable is used to make a date field available for the due date.
CurrentStepOverdueDate	Dynamic date: This variable is used to display the time (number of days) since the job or data object has been in a workflow step.
WorkflowOverdueDate	Dynamic date: This variable is used to display the number of days that the job or data object is overdue (in relation to the defined due date for the workflow step).
Job/data object name	Single inputline: This variable is used to provide an input field for the job or data object name.
JobIdFormatted	Constant variable: This variable is used to display the unique ID of the job or data object.
JobTypePseudoVariable	Job type: This variable is used to display the job type or data object type that is in use.

Technical Name	Variable type/Description
Comments	Chat desc from props: This variable is used to make the comment function available.
LastModificationDate	Datepicker: This variable is used to display the date of the last change.
SubJobs	Subjobs: This variable is used to make the functions required for sub-jobs or sub data objects available.
SystemPrice	Single inputline: This variable can be used to assign fixed or variable prices to the Service item types for the Marketing Shop.
default_media	Asset Selector: This variable is used to make the *Default media* variable available.
CurrentStepStartDate	Dynamic date: This variable is used to display the start date of the current workflow step.
JobState	Bean property: This variable is used to display the current workflow step.
TaskManager	Task Manager: This variable is used to make the functions for the Task Manager available. Note: Do not use in 8.0 anymore!
Themes	Description text: This variable can be used to assign one or more themes to the job or data object.
WorkflowObjectId	Workflow: This variable is used to display the workflow that is in use.
WorkflowStartDate	Dynamic date: This variable is used to display the start date of the workflow.

The Categories (*Themes*) system variable must be used at least once on a datasheet for Filter by *Category* to be available in the Job Overview.

6 Appendix

6.1 Tips: Roles and Permissions

Permission management via user roles for the Job Manager is an important component to provide administrators and users with different skills and functions or to explicitly prohibit and hide them in the user interface.

It is advisable to know exactly which rights to use and to test them thoroughly before putting the system into operation. This is because the internal designations of the rights do not always use meaningful names and sometimes only the combination of rights unlocks a function in the user interface. All permissions for the *Jobs* module are completely listed and described in Chapter 11.1.7 of the Administration manual. The permission for the *Data Hub* module can be found in chapter 11.1.8.

> Note
>
> The following previously used permission entries are still visible and selectable when editing a role, but their assignment to a role has no functional significance:
>
> - *Consume JMS Messages*
> - *Show Blueprint Jobs*

6.1.1.1 Example of a "Clerk" Role

The person can see and access all organizational units and their objects. The user may edit jobs in the workflow steps assigned to him and also create new jobs from the stored types/processes. They are not allowed to skip workflow steps. This user is not allowed to *complete*, *cancel*, *delete* jobs nor has access to the *Export* button or the *Total Filter* menu.

The *Advanced Search* is available (*Manage Filters*) and a *Public Filter* may be saved from it. The BPMN workflow tab is not active. In the *Workflow* tab for standard jobs, only the next step may be selected and none may be skipped. The person is allowed to start a review on the asset in the datasheet.

The role consists of these permissions:

- *Access All Orgs*
- *Create JM Request*
- *Edit JM Request*
- *Manage Filters*
- *Manage Review*
- *Manage Views*
- *Module Access*
- *Select Type*

6.1.1.2 *Example of an "Assistant" Role*

Here the rights are even more restricted. Persons with this role are only allowed to fill in the datasheet from jobs assigned to them and then forward them to the next workflow step. No new jobs can be created other than the preset default job type and the optional *Only Briefing*. The assigned persons are only empowered to create jobs from the preset default job type. They cannot select any other job types.

The role consists of these permissions:

- *Change JM Request*
- *Create JM Request*
- *Edit JM Request*
- *Module Access*

6.1.2 Important Permissions

See All (Finished, Deleted, Canceled) Jobs

Prerequisite

In order for an administrator in the *Jobs* module to be able to select and open all other jobs in the filter menu, their user role must have the *Act Like Creator* permission. This enables them to access running jobs/processes with the same permissions as the creator, provided this is not restricted in the type.

Advanced Search

Prerequisite: The user role has the *Manage Filters* permission so that the advanced search can be used.

Saving and publishing filters

Prerequisite

The user role has the *Publish Filter* permission.

Finishing a job

Prerequisite

The user role has the *Finish JM Request* permission.

Copying a job

What is meant here is the user command in the datasheet menu: ... > *Clone and not Copy type in the DSE administration.*

Prerequisite

The user role has the *Copy Job* and *Create JM Request* permissions. For users without this combination of rights, the menu command remains hidden.

Canceling a job

Prerequisite

The user role the *Close JM Request* permission.

Delete a job

Prerequisite

You are the creator of the job/process or your role in the *Jobs* module has been assigned the *Delete JM Request* permission.

Undo deletion

You can reactivate jobs that have been marked as deleted.

Prerequisite

You are the creator of the job/process or or your role in the *Jobs* module has been assigned the *Undelete* permission.

De-activate/reactivate

Prerequisite

The user role has the *De archive* permission.

Publish filter

Prerequisite

The user role has the *Manage Filters* and *Publish Filter* permissions.

Changing the creator of a job

Prerequisite

The user role has the the *Change JM Request* permission.

Customize column view

Makes the *Add column* dropdown menu available in the Job overview.

Prerequisite

The user role has the *Manage Views* permission.

Forwarding over multiple steps

Prerequisite

For classic jobs, the *Skip Workflow Steps* permission is required for the user role for forwarding over several steps.

Export

Prerequisite

You need the *Export XML* permission to initiate exports. For users without this permission, the *Export* button and the *Exports* page remain hidden. Without the *See all Exports authorization*, even administrators cannot see and download other users' exports. Since the *Export Jobs* command also stands in the *All Filters* dropdown menu, the XML export can currently only be completely disabled for those roles that do not have either the *Close JM Request* or *Delete JM Request* permission.

6.1.3 Resource Management

> Note
>
> In the current version 8.0 resource management is discontinued.

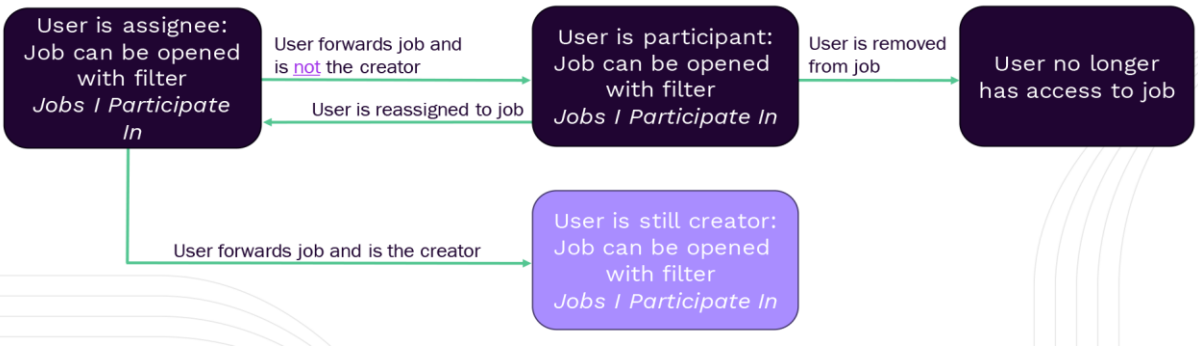
6.2 Former Assignees Automatically Added as Participants to Jobs

In some cases, users may want to access a job that they have participated in at a later date, even if they are no longer involved in it, by using the *Jobs I Participate In* filter on the Job Manager search page. Previously, when users without sufficient permissions attempted to reopen a job again after forwarding it, they would receive an error message that was not easy to understand.

Seamless addition as a participant ensures that former assignees retain visibility into jobs they previously worked on, while enforcing appropriate access control. This update delivers a consistent experience for users across all job types, including processes (BPMN-based jobs).

How it Works

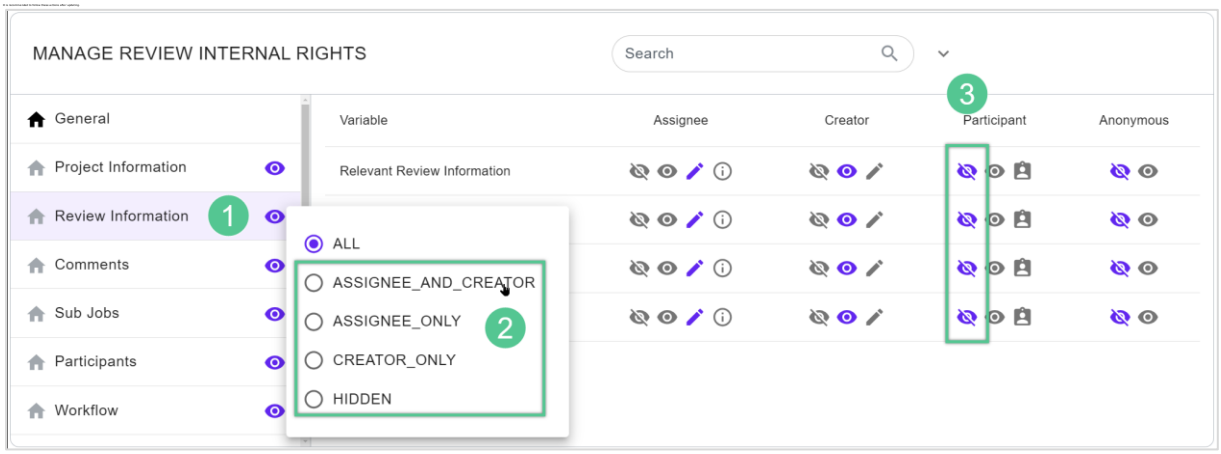
When a user forwards a job they are assigned to, the system will automatically add them as a participant in the background, with no user interaction required. The user will be listed under the *Participants* tab of the job. The job will remain accessible from their *Jobs I Participate In* filter. They can open and view the job, but with the access rights of a Participant role. If they later become the assignee again, they will revert to the *Assignee* role without an extra participant entry.



If they are also the creator, they will remain listed as creator without a participant entry. Participants can be manually removed, revoking their access. To avoid spamming users with notifications, no e-mail notification is sent when users are automatically added as participants in the background. This change does not affect Job Manager administrators and power users with "act like" permissions (e.g. *Act like Assignee*, *Admin JM Request*) as they always have full access.

> Example
>
> An agency that was an assignee in step 3 will also be able to open the job in step 7 with the limited access rights for a participant role. They will then see the datasheet and all its fields as configured for this role.

Post-Update Administrator Actions



> Note
>
> If there are certain fields or tabs that you want to remain invisible to mere participants in later phases, you will need to configure the participant access rights for the corresponding user tasks accordingly. Observe 1-3 in screenshot and the actions in the table below.

Action	Description	Reason
Update Access Rights Configuration	Ensure access rights are configured to restrict participant views to appropriate fields	To maintain data security and prevent unauthorized access to sensitive information
Review Job Filters	Verify if the Jobs I Participate In filter now includes jobs where the user was previously an assignee	To ensure that former assignees can see and access relevant jobs in their filter
Communicate Changes	Inform users about the change in functionality and its impact on job visibility and access	To ensure all users are aware of how the changes affect their workflow and access patterns

Discontinued in Assignee/Participant Context

Functionality	Description	Reason	Effective Version
Until version 7.4, former assignees are removed from participants.	User task and workflow assignees are no longer removed from the participants list after completing their task.	Filter Jobs I Participate In listed jobs that cannot be opened if user was assignee but no longer is.	8.0

Note

It is not possible to revert to the 7.4 assignee behavior in 8.0 and later.

6.3 Visibility by Management Level

6.3.1 Job View With Restriction by Organizational Unit

Organizing data based on the rights and roles system is not always easy. We will show you on the basis of a use case how to make it possible for supervisors to see only the jobs of their team members, and supervisors above them in the hierarchy to see the jobs of a correspondingly larger group of people.

Goal of the Configuration

The CMO wants to be able to see and edit their own jobs and also have read access to all the jobs that the marketing managers in their sub-departments have created. According to the organization chart, these are the positions of *Head of Marketing DACH* and *Head of Marketing Germany*.

Managers who do not head any other sub-departments are only allowed to see and edit the jobs that are in their area of responsibility.

Configuration Steps

You need a new modifiable structure with two attributes to uniquely assign users to an organizational unit.

1. Navigate to *Administration* > *Overview* > *Data Structures & Workflows* > *Custom Objects & Structures* > *Custom Structures*.
2. Type the name `PM_OrgUnit` into the *Add new structure* field.
3. Leave the selection menu at *Default*.
4. Click *Add*.

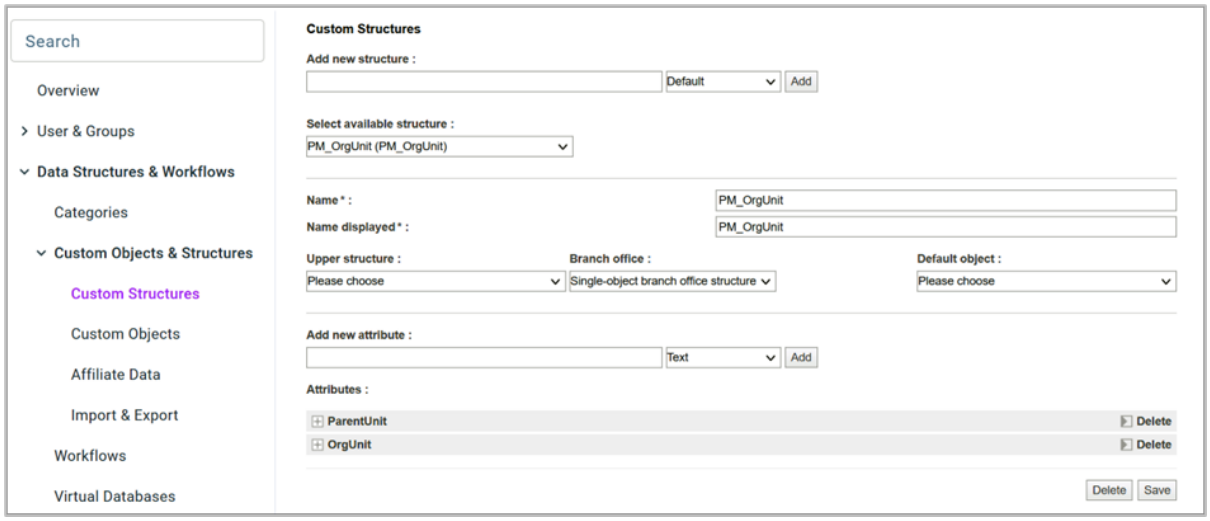
   You have created the `PM_OrgUnit` and can now add attributes in the bottom field named *Add new attribute*.

5. Type `OrgUnit` into the *Add new attribute* field, leave the selection menu on *Text* and click *Add*.

   You have created the `OrgUnit` attribute for the custom structure `PM_OrgUnit`.

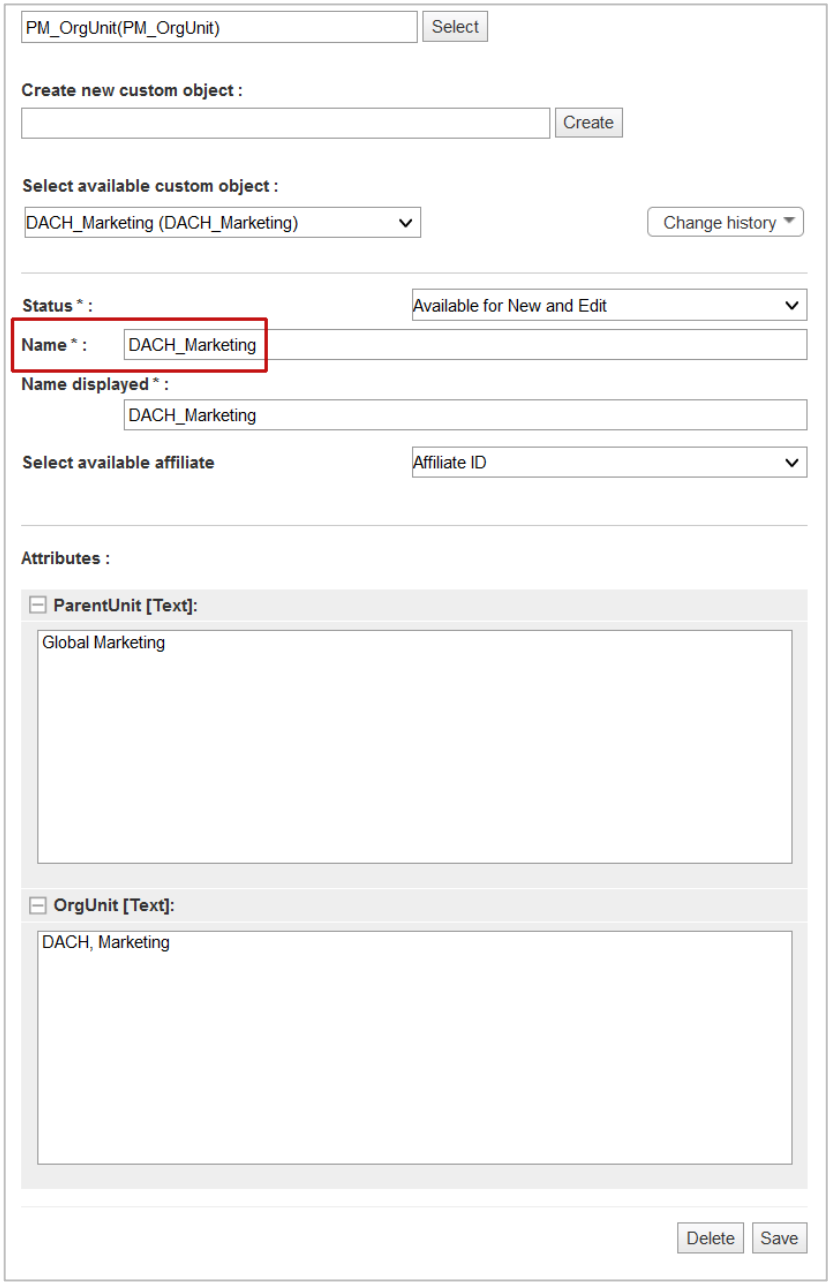
6. Repeat step 5 to create the `ParentUnit` as well. See illustration below the following note.

> **Note**
>
> In the majority of cases, the assignment of an organizational unit is sufficient. In order to map the hierarchical classification of the user within their organization, it is also useful to specify `ParentUnit`. In order to be able to filter even more finely, it is conceivable to use further attributes (e.g. for sub-departments, budget, cost center).



For each organizational unit, you must now create a custom object and fill in the attributes accordingly.

7. Navigate to *Administration* > *Overview* > *Data Structures & Workflows* > *Custom Objects & Structures* > *Custom Objects*.
8. Create the custom objects for the modifiable structure you just created.
9. Also fill in the `OrgUnit` and `ParentUnit` fields.



The technical name outlined in red above will be stored in all user accounts that belong to this organizational unit in the following step.

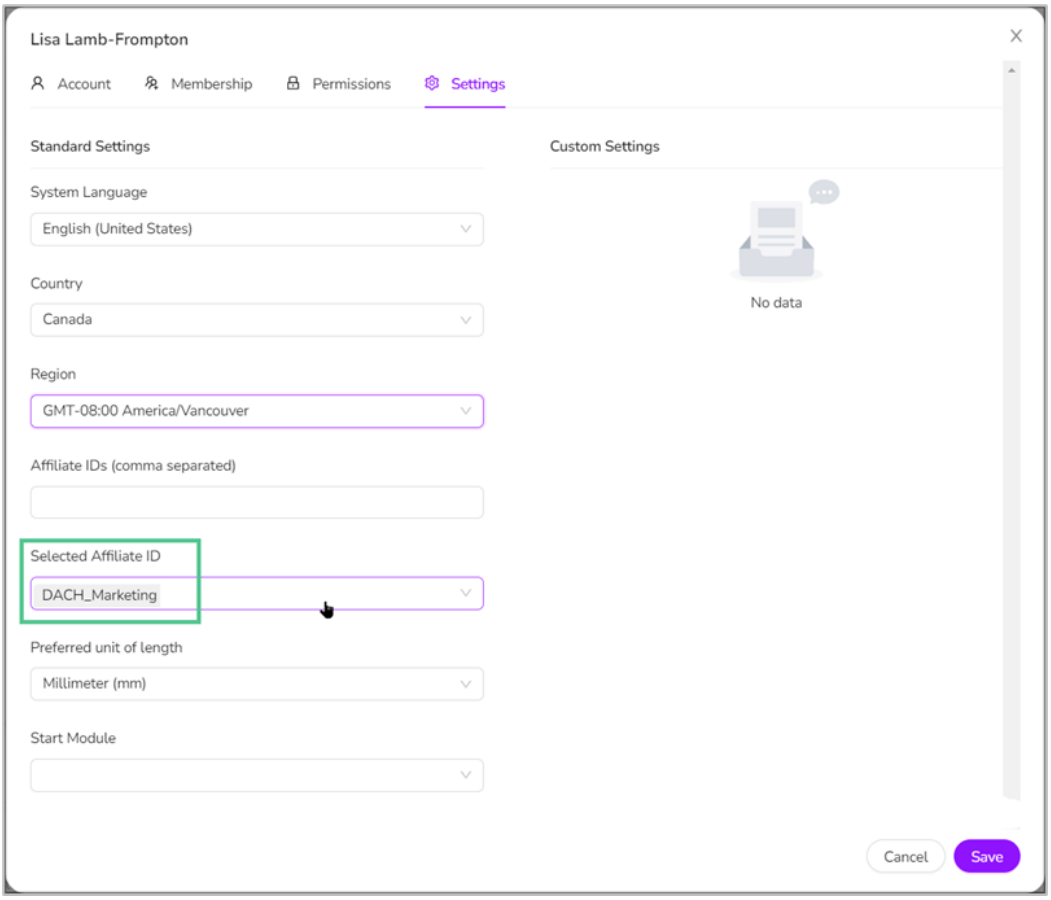
10. Copy the technical name from the name field to avoid input errors.
11. Navigate to > Administration > User List.
12. Search for a user to customize..
13. Click on the row of the user.

    A side panel is displayed on the right
14. Click *Edit account*.
15. Switch to the *Settings* tab.

16. Set the OrgUnit by pasting it from the clipboard into the field *Affiliate IDs (comma-separated)* and click *Save* to update the user account.

    After saving, the entry appears in the *Selected Affiliate ID* field.

17. Store this value for all users of the `DACH_Marketing` OrgUnit by repeating steps 12-16.



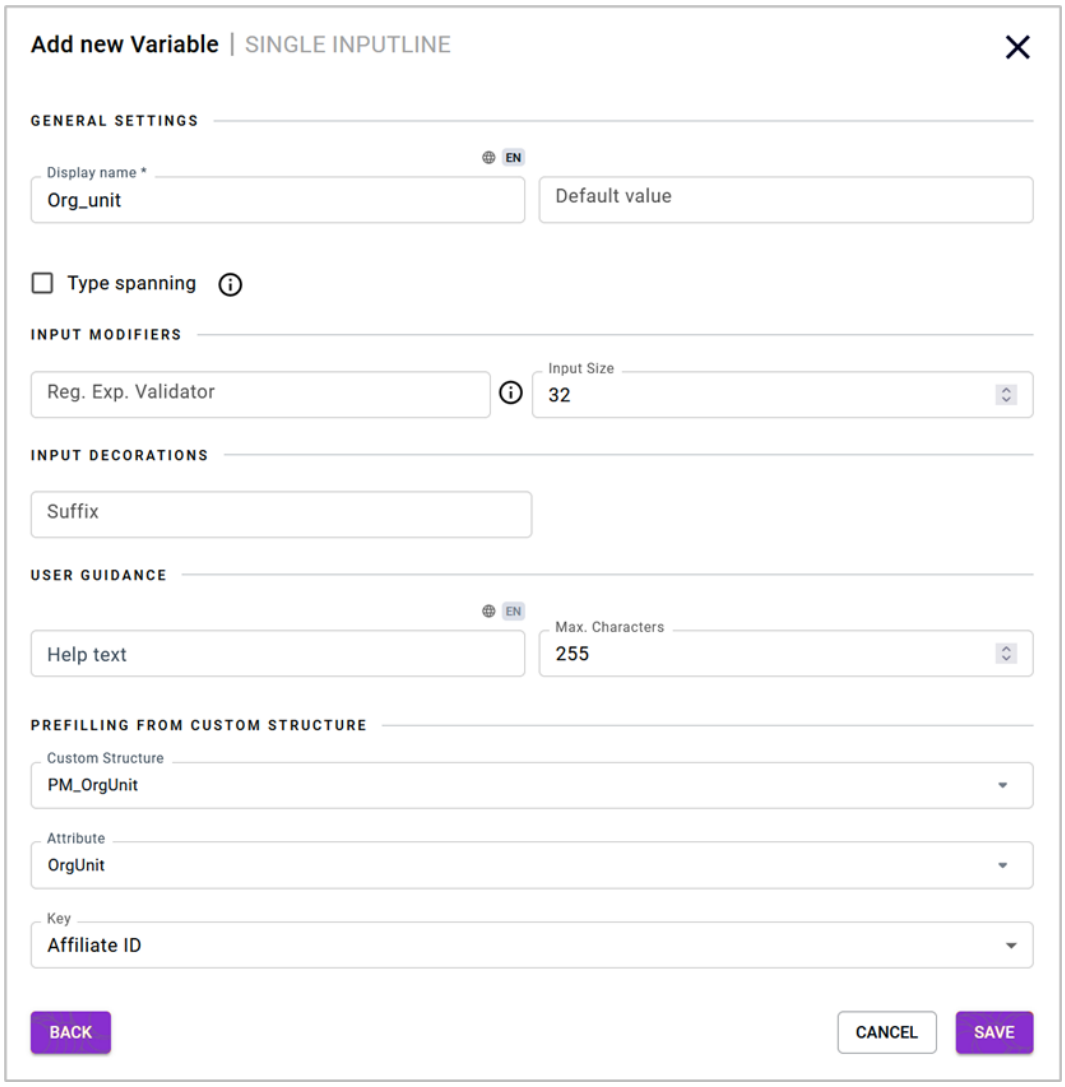
> **Note**
>
> The field content for *Selected Affiliate ID* (here `DACH_Marketing`) must correspond exactly to the technical name of the custom object previously created for this department, and must not be the displayed name (*Display name* * field).

    You have stored the affiliate ID in the form of the OrgUnit in the user account.

18. Add two fields to all relevant job types' datasheets. These will automatically populate the attributes from the custom object when a new job is created.

Customizing Datasheet

1. Navigate to > Administration > Overview > Datasheet Engine > Types.
2. Open an existing type that you want to customize or create a new one. See chapter 3.3.2 and following.
3. Create a variable (Type: *Single Inputline*) for the OrgUnit.
4. The variable name can be freely chosen, but when pre-populating from the modifiable structure, you must stick to the previously defined values.
5. Place the variable on the datasheet.

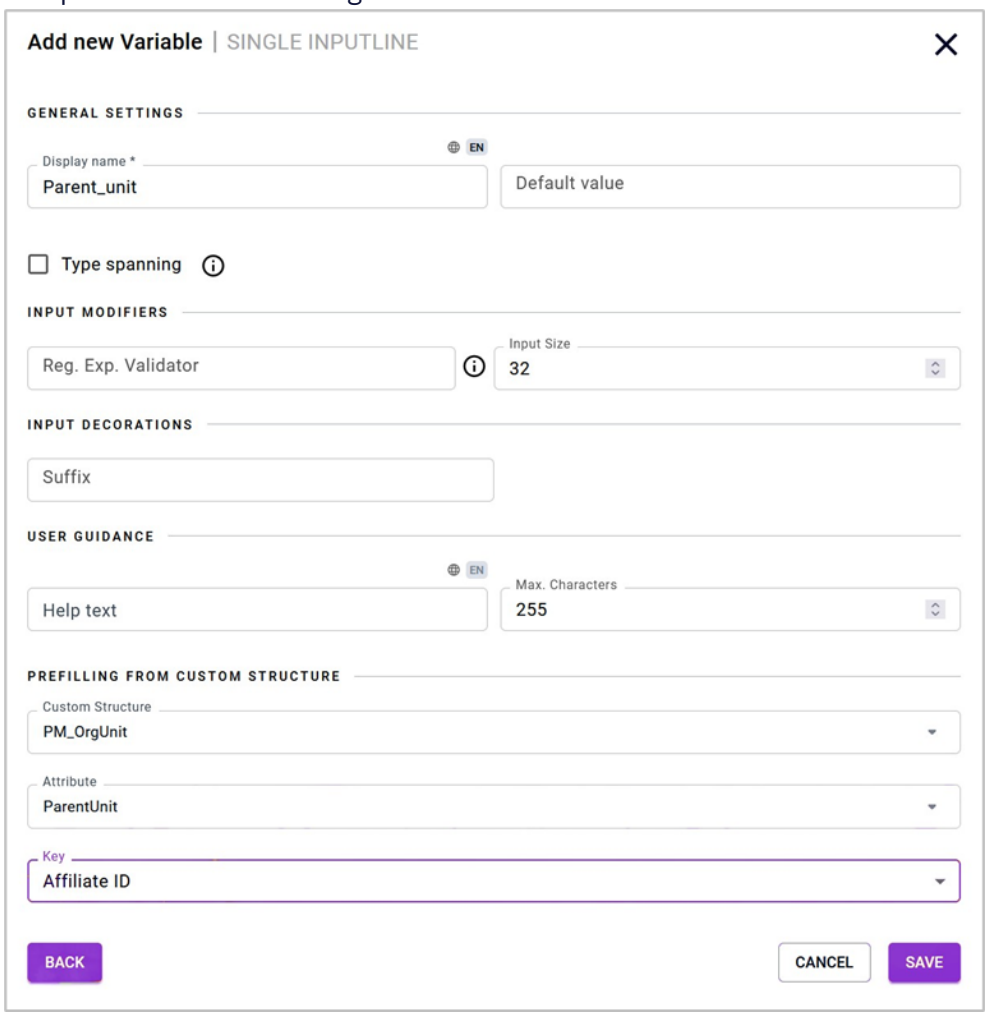


The figure above shows the variable for the OrgUnit.

6. Now create a variable (*Single Inputline*) for the ParentUnit of `DACH_Marketing` according to the same principle.

   The variable name can be freely chosen, but when pre-populating from the modifiable structure, you must stick to the previously defined values.

The following figure illustrates the process of creating the variable for the parent unit of the OrgUnit.

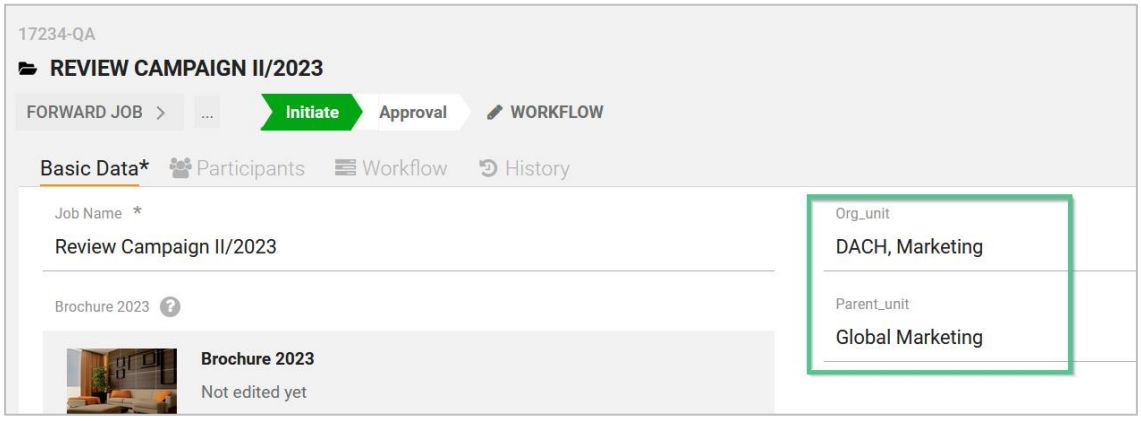


7. Publish the changes and create a new job based on this job type.

Each time a new job is created for the department head for whom a value has been stored in the *Selected Affiliate ID* field in the account, the open job datasheet will display the attribute values of their organizational unit and the ones above it.

Following the completion of the previous steps, it is now possible to clearly assign a job to a user and their organizational unit.

Example in Datasheet



6.3.2 Checking the Configuration

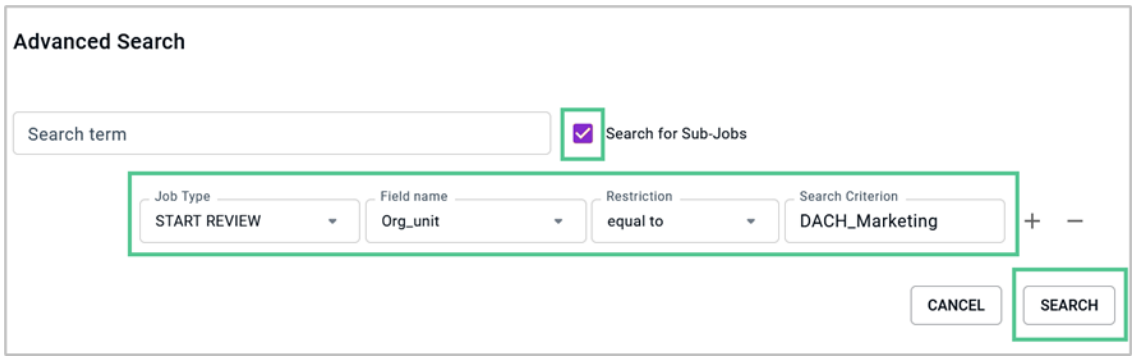
You need to make sure that a user can only see and edit their own jobs and those of their subordinate departments. For this purpose, a filter is set for this user, which displays only the jobs created by users from the subordinate organizational units.

Prerequisite

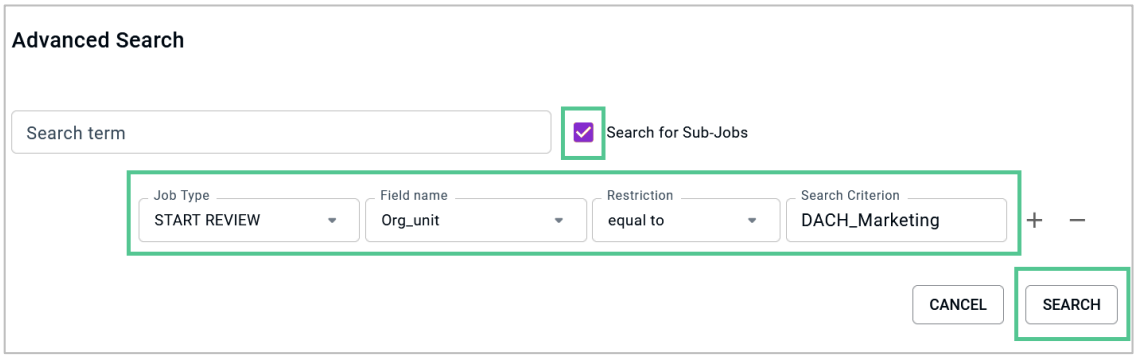
- To be able to set up this filter, the user role must initially have the *Manage Filters* permission.

> **Note**
>
> After creating and testing the filter, you must revoke the *Manage Filters* permission from the user again to prevent them from creating their own filters for jobs from other organizational units.

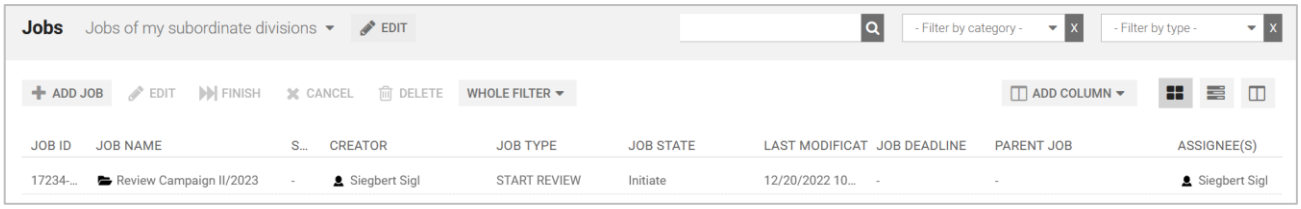
1. To create the filter, log in as this user.
2. On the Job Manager home page, start an *Advanced Search*, in the Filter menu and click *Edit*.



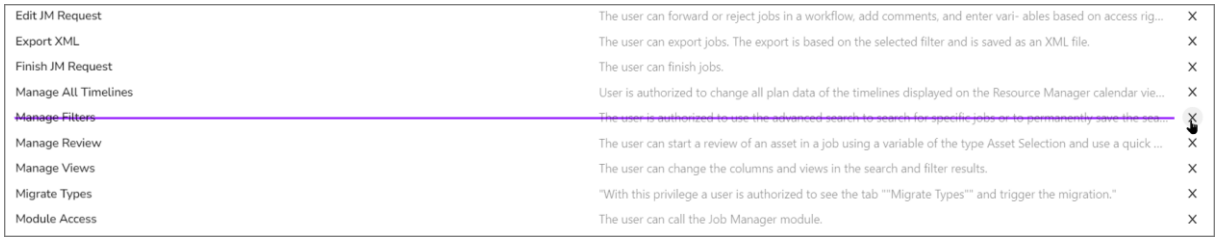
3. Fill in the search as shown in the figure above without a search term. The search criterion here is the displayed name and not the technical name as before, i.e., `DACH, Marketing`.
4. Click *Search*.
5. Set the checkbox at *Set as current filter and press Save*.



Desired result: The logged-in CMO now sees the appropriately prepared jobs of the subordinate organizational unit "DACH, Marketing" with this filter.



6. Optional: If you extend the filter shown above with further search strings with OR, which each contain the visible name of another organizational unit as a search criterion, then further jobs of other, subordinate organizational units will also be displayed, provided they have been created accordingly.
7. Remove *Manage Filters* permission from the role.



The role is now limited to ensuring that users who are affected by the change only see relevant jobs and organizational units. It is no longer possible for them to create their own filters.

