Workflows

Release 15.4.1.3

CONTACT Software

Oct 29, 2018

Contents

1 Introduction **1**

2 Designing a Workflow **2**
2.1 Creating Workflow Objects . . . 2
2.2 Workflow Designer . . . 3

3 Running Workflows **17**
3.1 Status Network of Workflows . . . 17
3.2 Processing Interactive Tasks . . . 20

4 Monitoring Workflows **21**
4.1 Protocol of a Workflow . . . 21
4.2 Email Notifications . . . 21

5 Permissions **25**
5.1 Workflow Owner . . . 25
5.2 Effective Permissions . . . 25

6 Examples **29**
6.1 Request for Quotation . . . 29
6.2 Release Part . . . 30
6.3 Engineering Change Order . . . 32

7 Glossary **33**

Index **38**

CHAPTER 1

Introduction

Note: Product development has high demands with respect to process support. Characteristic features are, for example, a high degree of parallelism and unforeseen changes and corrections in the course of a project. Conventional solutions soon reach their limits here. *CONTACT Workflows* offers comprehensive support for these requirements.

Processes in product development consist of a large number of partial goals and results, which cannot be described in detail in advance. In addition, the information is often of importance and has to be districuted beyond the development process as part of product lifecycle management.

CONTACT Workflows lets you define *tasks* which are to be executed within the scope of a higher-level *workflow*. The main feature of *CONTACT Workflows* is structuring, supporting and controlling the information flow between the involved parties using these *tasks*. Additionally, *system tasks* can be combined with interactive *tasks* (such as notifying *workflow* participants or cancelling a *workflow*).

The *Workflow Designer* coordinates data and resources so business goals depending on either can be achieved without delay. For optimal traceability, the system logs the entire *workflow* in order to meet documentary and legal requirements.

The specific components of *CONTACT Workflows* are

- a *Workflow Engine*, which runs *workflows*,
- the *Workflow Designer* to graphically design *workflows* and
- a plugin for *CONTACT Tasks*, which supports users in processing their *tasks*.

Further information about the *Workflow Engine* can be found in the Administration Manual.

CHAPTER 2

Designing a Workflow

2.1 Creating Workflow Objects

Usually you will instantiate a *workflow* from an existing *workflow* template in status  **Released**. However, it is also possible to freely design new *workflows*.

2.1.1 Instantiating Workflow Templates

Right-click the menu item *Processes -> Workflows* and select the *New from template* in the context menu. In the following dialog you can select a *workflow* template in status  **Released**.

Alternatively, you can also directly create a workflow from certain object classes such as Documents by clicking *Workflow -> New from Template* in their context menu. *Workflows* instantiated this way already contain the context object in their global *briefcase* "Attachments". This mode only lets you choose from templates specifically released for the context object's class (see *Rules for Selectable Workflow Templates* (page 2))

In any case, the newly instantiated *workflow* will then be opened in *Workflow Designer*.

Rules for Selectable Workflow Templates

Depending on the *tasks* defined in the template, the template may only be useful for certain objects. A template performing status changes, for example, can usually only be used for objects of certain classes or object lifecycles. Therefore, you can assign rules to a template in the tab *Object Rules*. A template is displayed in the template selection catalog if

- no rule is attached to it or
- at least one attached rule matches the object *Workflow -> New from Template* was called on.

2.1.2 Creating a Workflow from Scratch

Right-click the menu item *Processes -> Workflows* and select *New... '*. The following dialog lets you define the metadata of the new *workflow* object.

ID The system will automatically generate a unique ID after confirming this dialog.

Template If you want to create a reusable *workflow* template, check this option. The option is available only to users of the roles *cdbwf: Process Administrator* and *cdbwf: Process Library Manager*.

Title Enter a meaningful title for the *workflow*.

Category You can optionally assign a *workflow category* to the *workflow*. The available *workflow categories* are determined by your administrator.

Project No. Optionally select a project to assign the *workflow* to it.

Start The system will fill in the current date as soon as the status of the *workflow* is changed to **Execution**.

Deadline Here you select the date on which the *workflow* must be completed at the latest.

Max. Duration (Days) Here you can enter a value for the maximum duration of the *workflow*. The value is used for calculating the *Deadline* if no value is entered there. The value represents a number of working days. You can use this field to schedule *workflows* independent of weekends and public holidays.

Status When saving the *workflow*, its status is initialized with **New**. For more information on the status network, see *Status Network of Workflows* (page 17).

Responsible Denote a person or role responsible for the *workflow*. If you create this *workflow* as a template, select either the role *cdbwf: Process Administrator* or *cdbwf: Process Library Manager* to make sure members of the respective role can edit this *workflow* template. The person actually in charge of a *workflow* is determined when the *workflow* is instantiated. The selection of a responsible person also fills the field *Organization*.

Organization This field makes finding the appropriate person easier. Enter the organizational unit of the user you want to name responsible. You can then select the desired user from a filtered list including only users of the corresponding organizational unit.

Description May contain further details relevant for processing the *workflow*.

Click *New* to save the *workflow* and immediately open it in the *Workflow Designer*.

2.2 Workflow Designer

Note: General hints for using the *Workflow Designer*:

- Some texts can be edited after clicking them.
- Changed values are saved in the database as soon as you leave the respective field. If you want to undo a change, press the `Escape` key before leaving the field.

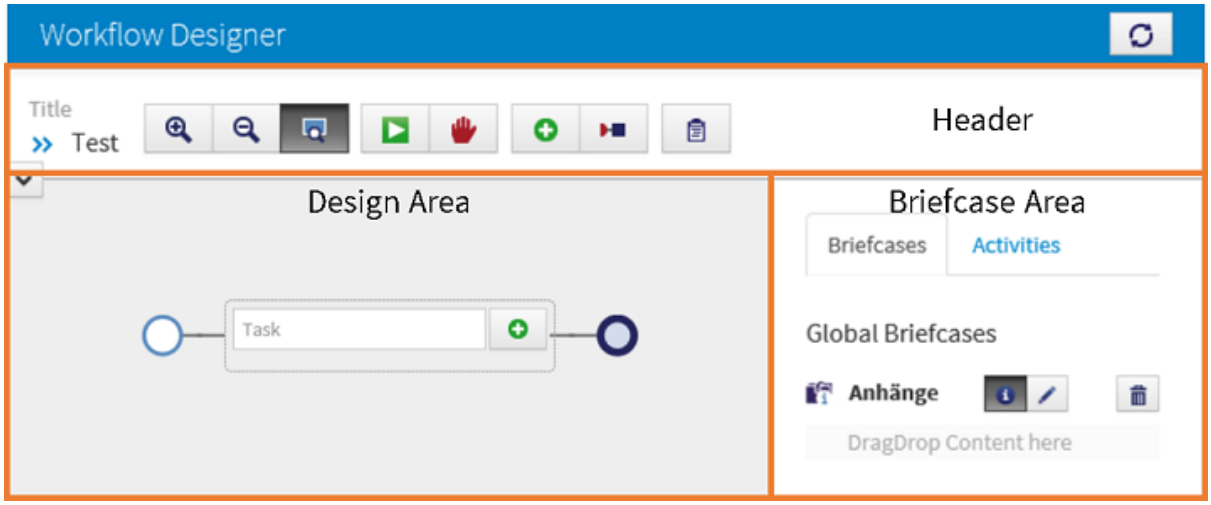


Fig. 2.1: The *Workflow Designer*

The *Workflow Designer* consists of the three main areas: The *header* (page 4), the *briefcase area* (page 4) and the *design area* (page 5).

2.2.1 Header

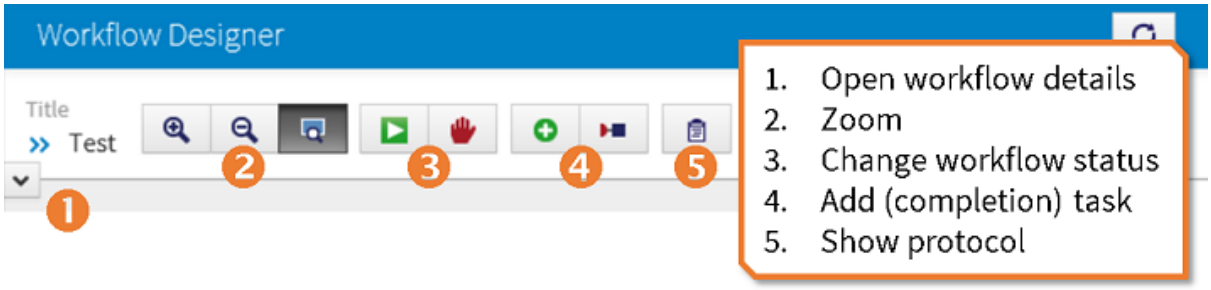


Fig. 2.2: The header of the *Workflow Designer*

The header contains the title of the currently opened *workflow* on the left. Below the title, a button opens the detail area (*Item 1* (page 4)). The detail area contains the *workflow's* project assignment and detailed description. These fields can be edited.

Next to the *workflow's* title is a toolbar:

- *Item 2* (page 4), Zoom: With these buttons you can zoom in and out of the design area or zoom to fit.
- *Item 3* (page 4), buttons for changing the *workflow's* status (see table below).
- *Item 4, left* (page 4) lets you add a *task*. This function usually requires the selection of existing *tasks* beforehand. This button makes the function visible and displays an appropriate message.
- *Item 4, right* (page 4) lets you add a *completion task*. The *completion task* is created outside of the regular *workflow* structure, i.e. after the completion symbol.

Workflow Status	Button	Description
New	Start Workflow	Not available for templates. Changes the *workflow's* status to **Execution** (*Item 3, left* (page 4)).
New	Discard Workflow	Changes the *workflow's* status to **Discarded** (*Item 3, right* (page 4)).
Execution	Freeze Workflow	Changes the *workflow's* status to **Frozen** (visible instead of *Item 3, left* (page 4) if the current status is **Execution**).
Execution	Cancel Workflow	Changes the *workflow's* status to **Failed**.

2.2.2 Briefcase Area

Briefcases

The briefcase area allows you to link *tasks* to objects such as documents. Linking is done via either global or local *briefcases*. Global *briefcases* are automatically available to all interactive *tasks*. Local *briefcases* can be assigned to individual *tasks* in their detail area. This allows you to link content to selected *tasks* only.

To fill a *briefcase* with content, drag and drop an object from a table to the desired *briefcase*.

Activities

The activities area shows conversations about the *workflow*. See the application manual of *CONTACT Activities* for further information.

2.2.3 Design Area

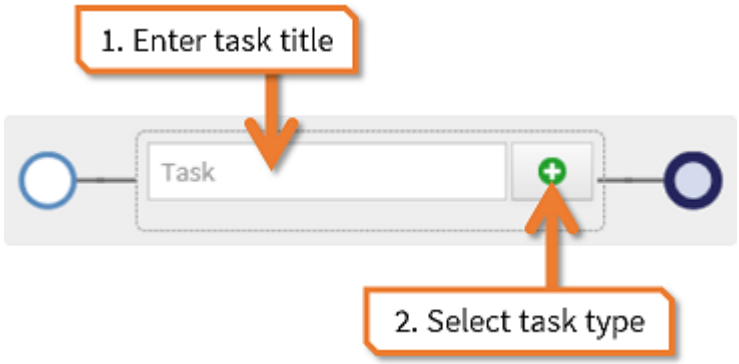
You can actually design your *workflow* in this area by creating *tasks* and putting them in temporal relation with each other (position them in parallel or sequential groups). Thus, highly complex *workflows* can be created.

Tasks are displayed as rectangles. The initial view only includes the most important information. To see all the details, you can use the small arrow at the bottom of a *task* to expand it.

Note: Time runs from left to right in the *Workflow Designer*: - Tasks displayed next to each other run in succession, while - tasks displayed one above another run in parallel.

Creating Tasks

Creating the First Task



A new *workflow* is represented in the *Workflow Designer* by a start and end point. Between these two points there is a placeholder text field for the first *task*.

To create a first *task*, enter a title in the text field, confirm with the `Enter` key and select one of the possible Task Types in the following dialog.

Task Types

Interactive Tasks

An interactive *tasks* in status **Execution** will block further processing of the *workflow* until one of their responsible users changes its status.

Note: *CONTACT Tasks* offers users convenient access to all of their *tasks* in status **Execution**.

Users responsible for a *task* can perform one of the following status changes, depending on its type:

Completed Completes the *task*, indicating success or approval,

Rejected Completes the *task*, indicating failure or rejection, and

Discarded Cancels the *task*, if the responsible user deems it irrelevant or incorrect.

The following table shows available interactive *task* types and

- potential statuses following **Execution** and their meaning,
- if a status change has to include a mandatory comment by the user, and
- if a status change to **Rejected** prematurely changes the status of the *workflow* to **Failed** or not.

Icon	Task Type	**Completed**	**Rejected**	Requires Comment	Prematurely Cancels Workflow
	Examination	Passed review	Failed review	Yes	No
	Approval	Approval granted	Approval denied	Yes	Yes
	Execution	Done	*not available*	No	No

Note: The status change to **Discarded** is available for all *task* types.

Note: The separate package *CONTACT Digital Signatures* contains a variant of the type Approval requiring the digital signature of a responsible user.

System Tasks

System tasks do not require a responsible user. *System tasks* in status **Execution** are run by the *workflow service*. Logically, the *workflow owner* is the one in charge. They vouch for the correctness of the design of the *workflow* with their name.

The following section uses "processing" synonymously for the processing of a *system task* in status **Execution** by the *workflow service*.

Information

Information *system tasks* serve the purpose of informing a select group of recipients of something at a certain point in time.

The recipient may be either a user or a role.

If the recipient is a role, only members of the role are informed at the time of processing. Subsequent changes to role memberships do not affect the *system task*.

Information *system tasks* are visible in *CONTACT Tasks* regardless of the status of the *workflow*, especially after its termination.

Copy

Copy *system tasks* copy all objects in their *briefcases* with *edit mode* **Info**. The copies are automatically added to the first *briefcase* assigned to the *system task* with *edit mode* **Edit**.

A common use case is to create a backup copy before an object is edited in the later stages of the *workflow*.

> **Warning:** Copy *system tasks* may only have exactly one *briefcase* with *edit mode* **Edit**.

Run Operation

This *system task* executes an arbitrary operation, the input for which is provided by *forms*.

All *forms* in *briefcases* of the *system task* are evaluated and the resulting attribute/value pairs are used to execute the operation defined in the *system task*.

> **Warning:** *Workflow* owners must make sure there is no unwanted overwriting of values when using more than one *form*. *Forms* in *briefcases* of *edit mode* **Edit** overwrite values of the same names found in *briefcases* of *edit mode* **Info**, but the order within these two groups is not deterministic.

If at least one operation configuration of applicability `Meta` exists for the operation, the operation is performed once without context.

For each other operation configuration, the operation is performed as follows:

- Once for each unique class name within the *briefcase contents* of **Edit** *briefcases* (except *forms*). This requires an existing operation configuration for the respective class name and applicability `Class`.

- Once for each object of the *briefcase contents* of **Edit** *briefcases* (except *forms*). The prerequisite is an existing operation configuration for the respective class name and applicability other than `Meta` and `Class`.

Status Change

System tasks of this type change the status of objects when being processed. Both *briefcase contents* with *edit mode* **Info** and **Edit** are taken into account. However, the *briefcase contents* to be changed can be filtered using an optional filter rule. Filter rules are object rules.

Target statuses have to be defined numerically in the parameters of the *system task*.

Note: If it is impossible to change the status of one of the objects, a *protocol* entry will be written. The system then tries to change the status of the remaining objects.

If at least one status change has failed, the status of the *system task* will be changed to **Discarded**.

Note: If no objects remain after filtering, the task is set to status **Completed**.

Workflow Cancelation

Processing a *system task* of this type results in an immediate change of the *workflow's* status to **Failed**. Usually, this is used in combination with a *constraint* to terminate the *workflow* only under certain conditions.

Note: A conceivable example would be to continue a *workflow* only if all objects within a *briefcase* are in a certain status.

Task Details

To view and edit the details of a *task* in the *Workflow Designer*, click on the button at the bottom right of the *task*.

This chapter describes the upper area of the details. Other areas are described in separate chapters:

- *Constraints* (page 11)
- *Forms* (page 13)
- *Briefcases* (page 15)

Setting a Deadline

You can set a deadline for the *task* to be completed. Current *tasks* that are overdue will be highlighted in *CONTACT Tasks* accordingly.

Click on the calendar icon and select the desired day.

Maximum Duration

Instead of a specific deadline, you can also specify a maximum duration in working days. This is then converted into a specific deadline when the status of the *task* is changed to **Execution**.

Complete Prematurely

Note: The option *Complete Prematurely* is deactivated by default. Please ask your system administrator to activate it.

Note: The option *Complete Prematurely* is only available for *tasks* of the type *Approval* and *Examination*.

With this option you can define a *task* within your *task group* as "master". This means that changing the status of this *task* to **Completed** will also set the parent *task group* to **Completed**. The other *tasks* of the same *task group* with status **Execution** will be changed to status **Discarded** at the same time.

Example *Complete Prematurely*

Given there are two users authorized for an approval, and the approval of one of them is sufficient to continue the process.

This can be modelled by defining two parallel *tasks* of type *Approval*. For both *tasks*, the option *Complete Prematurely* is activated.

In *CONTACT Tasks*, each of the two responsible users now has the opportunity to

- approve their *task* by changing its status to **Completed** or
- deny their approval by changing the status of the *task* to **Rejected**.

Any decision has to be explained in the status change's comment. The status change also sets the status of the parallel *task* to **Discarded** and the status of the parent *task group* to **Completed**. Effectively, the first decision counts.

Uses Global Briefcases

Note: Only *system tasks* have the option *Uses global briefcases*.

Because *system tasks* usually require more specific input data than interactive *tasks*, they do *not* use global *briefcases* automatically.

If global *briefcases* are to be used by a *system task*, the option *Uses global briefcases* can be activated for the *system task*.

Defining Responsibility

For interactive Task Types you have to select a responsible user or role for this *task*. Click on the hand symbol below the task type of a *task*.

In the dialog box for selecting the responsible person, the top row contains a field in which you can select a role. The main area then only displays the users who are members of this role.

Below the main area is a row of fields in which you can enter filter criteria for the individual attributes. The fields for the filter criteria can be freely combined.

If no one is responsible for a *task* yet, you can select either a role or an individual user. For non-template *workflows*, it is advisable to prefer individual users to a role in order to define responsibility as clearly as possible.

For *tasks* within a *workflow* template, it is recommended to use roles instead. *Workflows* instantiated from such a template can then be more specific.

If a *task* already has a role assigned to it, the role is automatically preselected as a filter in the dialog for selecting the responsible person. In this case, you will only see the members of the role in the main area. To display all roles and users, select "-" in the filter field.

Creating Subsequent Tasks

After you have created the first *task*, you have to then define the position of any subsequent *task* before actually creating it. Use the mouse to drag a selection rectangle containing a set of previously defined *tasks*.

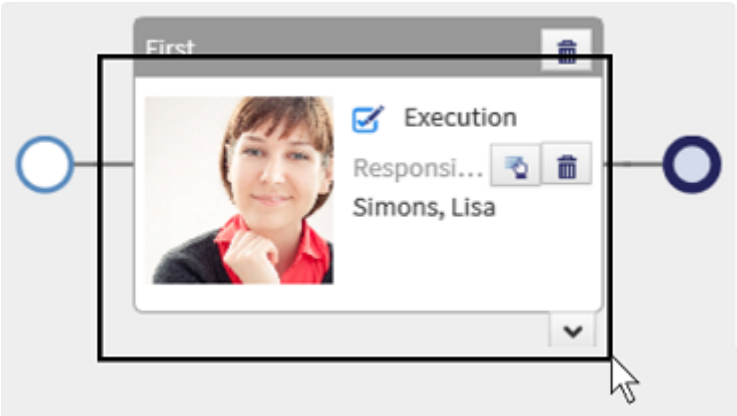


Fig. 2.3: Drawing a rectangle around one or more existing *tasks*

Upon releasing the mouse button, positions at which you can create a new *task* are marked with plus buttons. Click on one of the plus buttons to create a new *task* at this position.

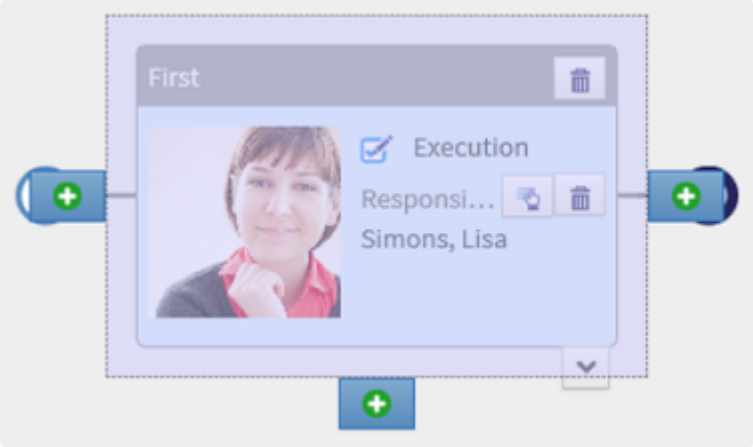


Fig. 2.4: Buttons for creating a *task* at each possible location

Note: **Reminder:** Time runs from left to right in the *Workflow Designer*: - Tasks displayed next to each other run in succession, while - tasks displayed one above another run in parallel.

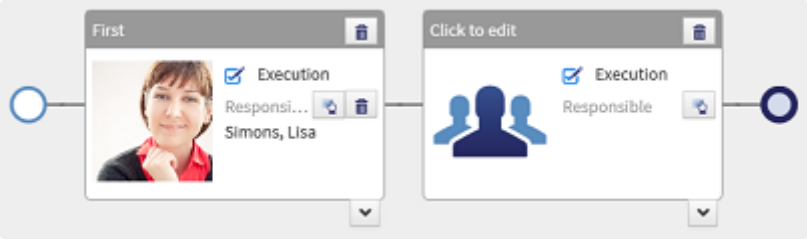


Fig. 2.5: Successor after clicking the right plus button

Note: You can change the title of a *task* later by clicking on the title bar of the *task*.

Note: Parts of the *workflow* that are already running or have already run (i.e. *tasks* with the status **Execution**, **Completed**, **Rejected** or **Discarded**) can no longer be changed. You can still add subsequent *tasks* as long as the status of the *workflow* is still **Execution**, though.

Changing Task Structure

To change the position of a *task*, you have to delete the *task* and recreate it at the correct position.

Constraints

All *constraints* of a *task* must be fulfilled immediately when the status of the *task* is changed to **Execution**.

If at least one *constraint* is violated, the *task* will be set to **Discarded** immediately.

Normally, *constraints* are checked against the *task*. If the *constraint* is assigned to a *briefcase*, it is checked against this *briefcase* instead.

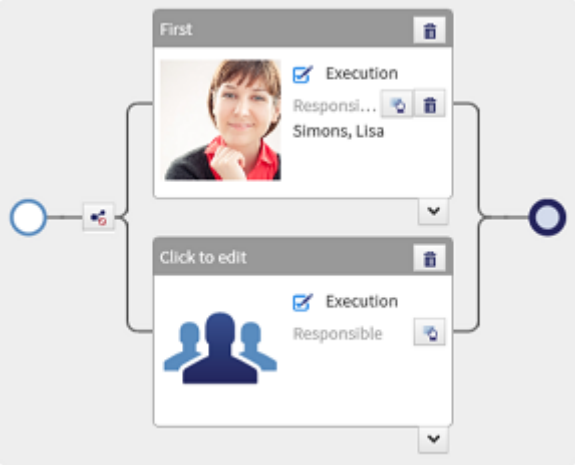


Fig. 2.6: Parallel *task* after clicking the bottom plus button

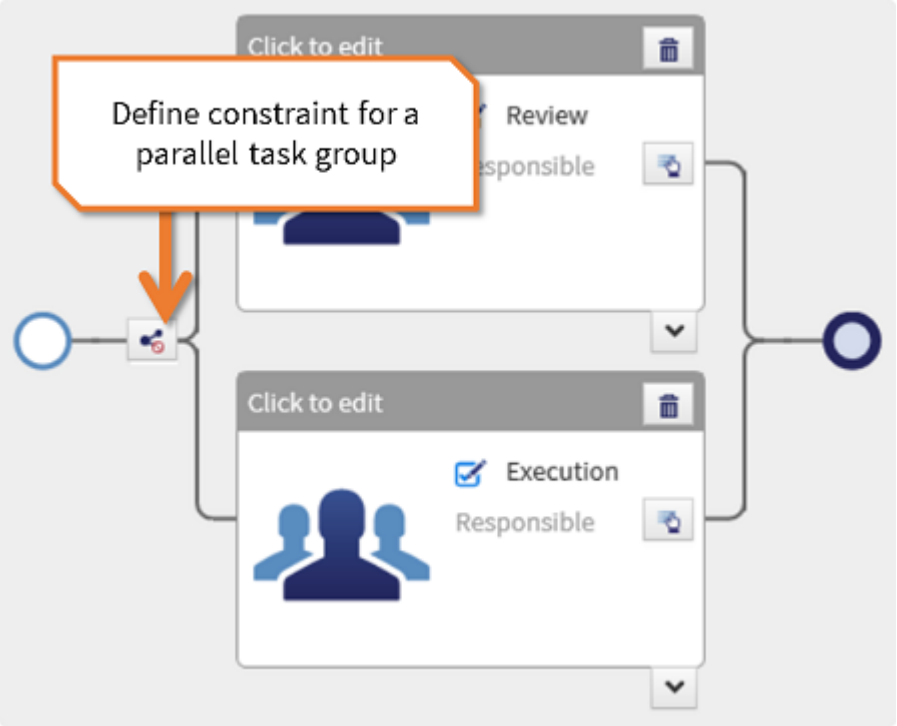
Defining Constraints for Single Tasks

In the details are of a *task*, you can specify one or more *constraints*. An example use case could be: "All objects of a folder must be released". Select the appropriate *constraint* and the corresponding *briefcase*. Only if all objects of the *briefcase* are actually released when the *task* is started, the *task* is considered relevant and remains in status **Execution**.

You can invert the *constraint* to make the *task* in the above example run only if not all objects are released, i.e. if at least one object's status differs.

The *constraints* of a *task* are connected logically by the operator "AND". This means that *all constraints* must be fulfilled for the *task* to be executed.

Defining Constraints for Task Groups



A *constraint* can also be defined for an entire *task group*. To do so, click on the *constraint* symbol at the branch that marks the beginning of the *task group*. Defining *constraints* will then be the same as for *individual tasks* (page 12).

Included Constraint Rules

The following table shows all included Objektregeln for the definition of *constraints*. Not listed are rules for other purposes, such as the representation in *CONTACT Tasks* or filter rules.

Rule	Description	Checked Against
wf-designer: previous process component done	Fulfilled for *tasks* whose predecessor's status is **Completed**. The predecessor is either the *task* with the next smaller position of the parent element or the predecessor of the parent element (if available). If no predecessor exists, or if it has a different status, the rule is violated.	*Task*
wf-designer: all briefcase objects approved	Fulfilled if all *briefcases* to be checked do not contain objects that are not released. By default, only documents and items are considered. Additional objects can be set up by specifying additional predicates.	*Task*, *Briefcase*
wf-designer: all previous tasks done	Fulfilled if all predecessors (all children of a previous parallel *task group* are considered) are in status **Completed**. Always violated for *briefcases*.	*Task*
wf-designer: all process tasks have start date and deadline	Fulfilled for a *workflow* if all of its *tasks* (not *task groups*) have non-empty attributes `start_date` and `deadline`.	*Workflow*
wf-designer: briefcase contains exactly 1 object	Fulfilled if the *briefcase* contains *exactly* one object.	*Briefcase*
wf-designer: parent's previous process component done	Fulfilled if the status of the predecessor of the parent *task group* is **Completed**. Violated if there is no parent *task group* or if it has no predecessor.	*Task*
wf-designer: process completing successfully	Fulfilled if the *workflow* is not currently being cancelled.	*Task*, *Briefcase*

Note: Your system administrator may define additional rules.

Forms

In the details are of an interactive *task* (examination, approval, execution) in the *Workflow Designer*, you can select a released form template. This instantiates a new *form* from the template and creates a local *briefcase* linked to the *task*, which also contains the new *form* itself.

Tasks can contain any number of *forms*. Only *forms* which are assigned to a *briefcase* of *edit mode* **Edit** can actually be filled in. *Forms* in other *briefcases* are displayed as read-only information.

Prefilling Form Data

As soon as at least one *form* is assigned to a *task*, it can be opened in a separate tab.

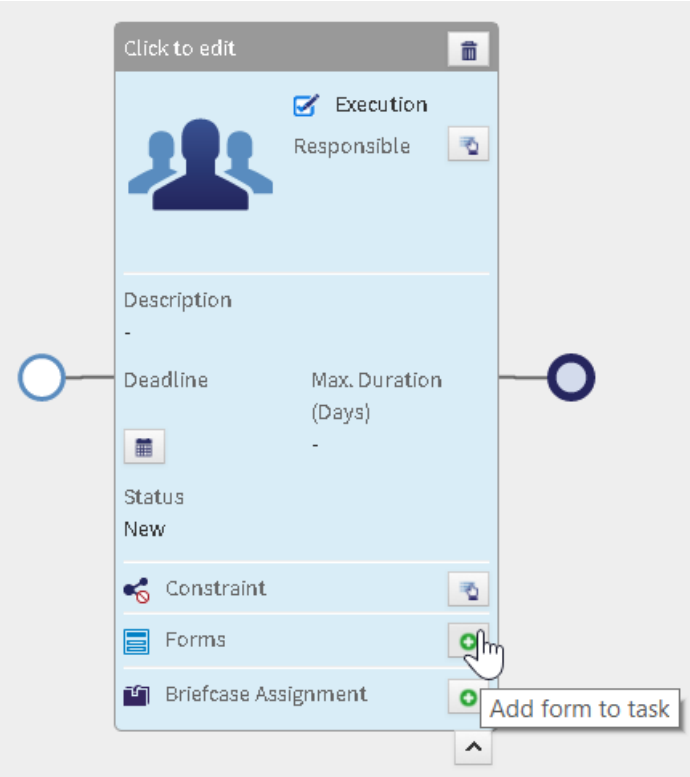


Fig. 2.7: Adding a *form* to a *task*

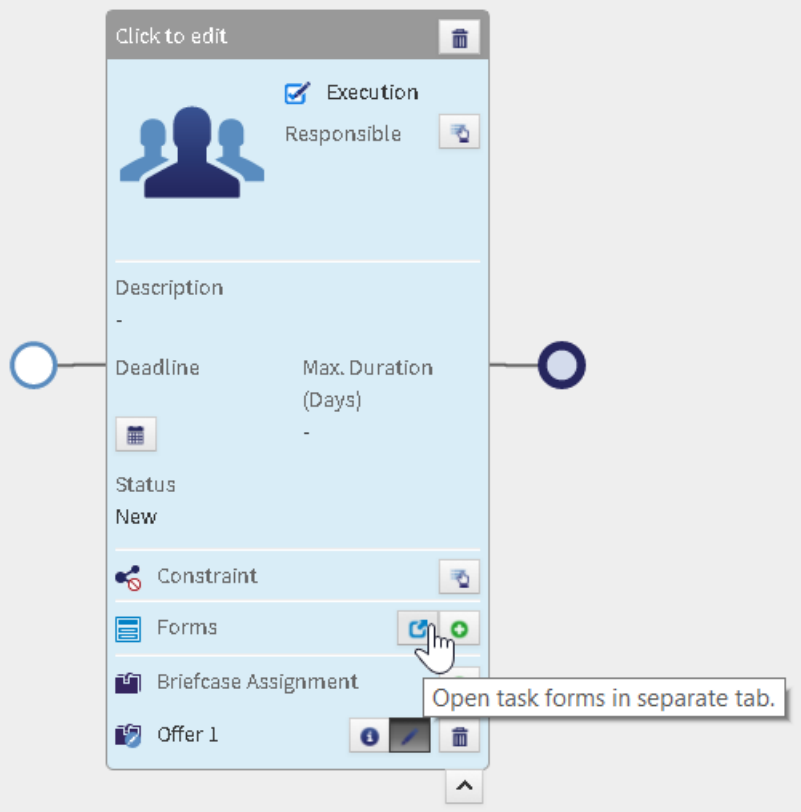


Fig. 2.8: Opening a *task's* the *forms*

Users responsible for a *workflow* can define default values in the *form* or assign further objects to the *briefcase* assigned to the *form*. These will prefill the *form* as soon as the *task's* status is set to **Execution**. Values entered manually are not overwritten.

Reusing Forms Across the Entire Workflow

The *briefcase* containing the *form* can be reused in subsequent *tasks*, for example to check its data or as input for *system tasks* of the type *Run Operation* (page 7).

Note: Before automatically processing *form* data by *system tasks*, it is recommended to have the data approved manually by an owner of the *workflow*.

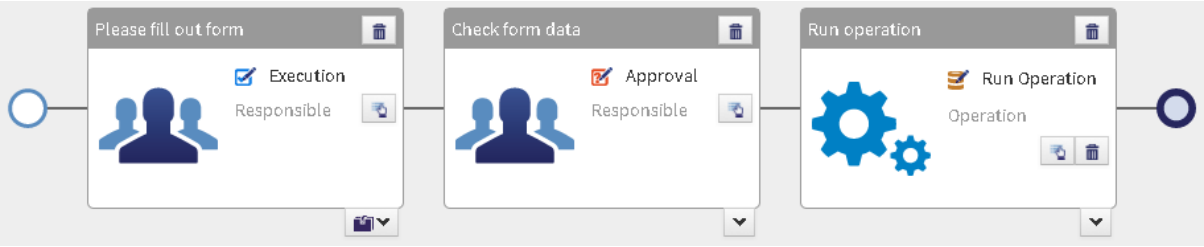


Fig. 2.9: Example concatenation of *form*, permission and *system task* "Run Operation".

Editing Forms

As soon as a *task* containing *forms* is set to status **Execution**, the person responsible can see the *forms* in *CONTACT Tasks* or *CONTACT Collaboration Portal* and - if assigned to a *briefcase* of *edit mode* **Edit** - edit their data.

Such writable *forms* can be edited and catalog values can be selected. In order to save the entered data, the user has to confirm by clicking the button `Submit Values`.

Tasks with *forms* can only be completed if all mandatory fields of the respective *form* are filled in.

Briefcases

Briefcases may either be *Global Briefcases* (page 15) or *Local Briefcases* (page 16). The reference of a *briefcase* to a *workflow* or *task* also is of a certain *edit mode*.

Briefcases of *edit mode* **Info** provide objects for viewing. Their *briefcase contents* can only be edited if the respective user is granted the necessary permissions by other means.

Briefcases of *edit mode* **Edit** provide objects for editing. The necessary rights to *briefcase contents* are "inherited" automatically. (see *Automatic Permissions* (page 16)).

Global Briefcases

Global *briefcases* are automatically used by every interactive *task* of the current *workflow*. The *edit mode* of global *briefcases* is defined directly when creating it. You have the options **Info** and **Edit**.

Local Briefcases

Local *briefcases* must be explicitly assigned to a *task* in order to affect the *task*. You first create local *briefcases* without *task* references in the briefcase area (right-hand side in the *Workflow Designer*).

It's not until you assign it to a *task* that you decide which *edit mode* the *briefcase* should be within that *task*. This means that the same local *briefcase* can be assigned to some *tasks* using *edit mode* **Info** and to others using *edit mode* **Edit**.

Note: The effect of the *edit mode* is the same for global *briefcases* and local *briefcases*.

Automatic Permissions

The system automatically grants the *workflow* participants the permissions necessary for processing their *tasks* in status **Execution**. Access is only granted via this mechanism while the respective *task* is in status **Execution** and the *briefcase* link's "Assign Access Rights" checkbox is checked.

Note: The checkbox "Assign Access Rights" is automatically checked as soon as a *task's* status is set to **Execution** and the *workflow owner* is granted the permissions themselves.

Permission may still be granted to *workflow* participants regardless of the *briefcases*.

Hint: Further details can be found in the Administrator's Manual.

Example *Automatic Permissions*

A *workflow owner* wants to grant write permissions on a confidential document for a specific purpose. They add the document to a *briefcase* in their *workflow* and assign the *briefcase* with *edit mode* **Edit** to a *task* for the responsible person.

If the *workflow owner* are granted write permissions on the confidential document themselves, the following applies: As long as the *task* remains in status **Execution**, any user responsible for it can change the document.

Note: If a *workflow* is running, the title bars of its *tasks* are be colored to reflect their statuses (see also *Status Network of a Workflow Task* (page 18)).

CHAPTER 3

Running Workflows

3.1 Status Network of Workflows

3.1.1 Status Network of a Workflow Template

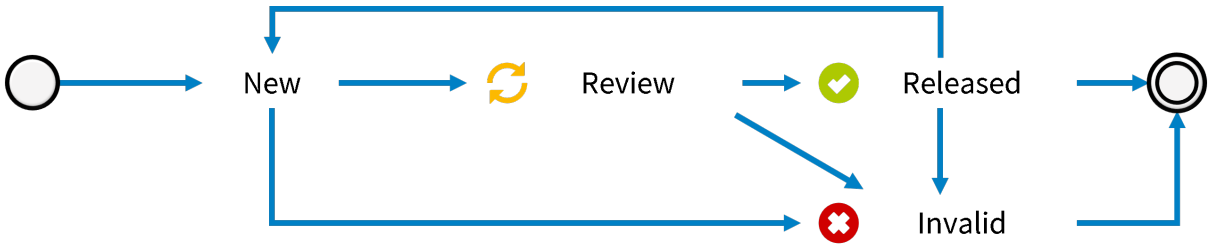


Fig. 3.1: The status network of a *workflow* template

Existing *workflow* templates can only be changed by members of the roles *cdbwf: Process Administrator* or *cdbwf: Process Library Manager*.

Status	Description	Editable?	Usable?
New	Initial status. The template is currently a draft.	Yes	No
Review	The template is currently being reviewed for a potential release.	No	No
Released	The template is released for usage.	No	Yes
Invalid	The template is obsolete and not usable anymore.	No	No

You can change the status of a *workflow* template using its context menu. Click the operation *Status change* in the template's context menu.

3.1.2 Status Network of a Workflow

The status of a *workflow* is only changed by the system or dedicated operations. The operation *Status change* may not be used. Instead, you can find the relevant operations directly in the *Workflow Designer* or context menu of a *workflow*.

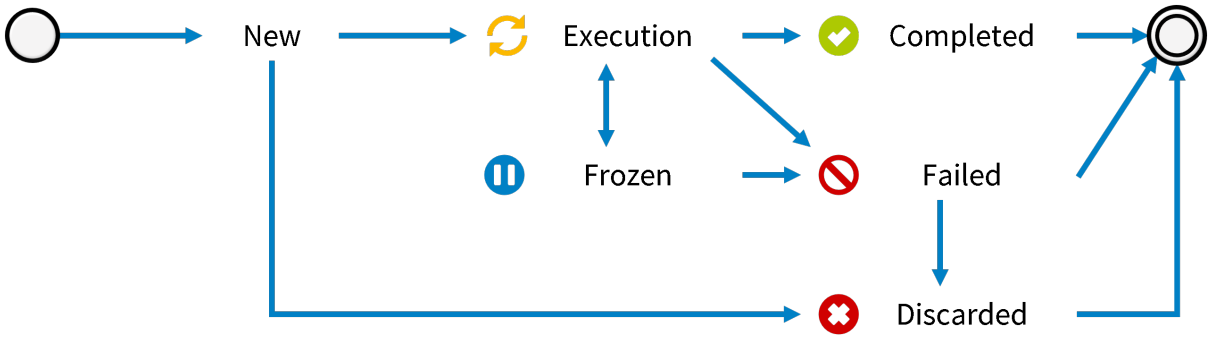


Fig. 3.2: The status network of a *workflow*

Status	Description	Editable?	Running?
New	Initial Status. The *workflow* is currently a draft.	Yes	No
Execution	The *workflow* is currently running.	Partly	Yes
Completed	The *workflow* has been terminated successfully.	No	No
Failed	The *workflow* has been terminated with a failure.	No	No
Discarded	The *workflow* was prematurely canceled.	No	No
Frozen	The *workflow* is currently paused.	Partly	No

Workflows automatically change the status of their *tasks* to **Execution** at the appropriate time. They respect the *task* structure defined in the *Workflow Designer*.

If a *workflow* is frozen, the status of all of its *tasks* will remain unchanged. However, the *tasks* are not shown in *CONTACT Tasks* while the *workflow* is in a status other than **Execution**.

If a *workflow* is stopped, it changes the status of its *tasks* in **New** and **Execution** to **Discarded** at the same time.

Note: Frozen *workflows* do not offer the following operations:

- Modify *workflow*,
- change the status of its *tasks*,
- modify, create, delete or copy its *tasks* and other objects of the *workflow* (such as *briefcases*, *constraints*, ...), and
- add or remove *briefcase contents*.

3.1.3 Status Network of a Workflow Task

The status of a *task* is mostly managed by the system (e.g. its *workflow*). Interactive *tasks* in status **Execution** are the exception: Their status is changed manually by one of their responsible persons as soon as they finish processing it (either successfully or not).

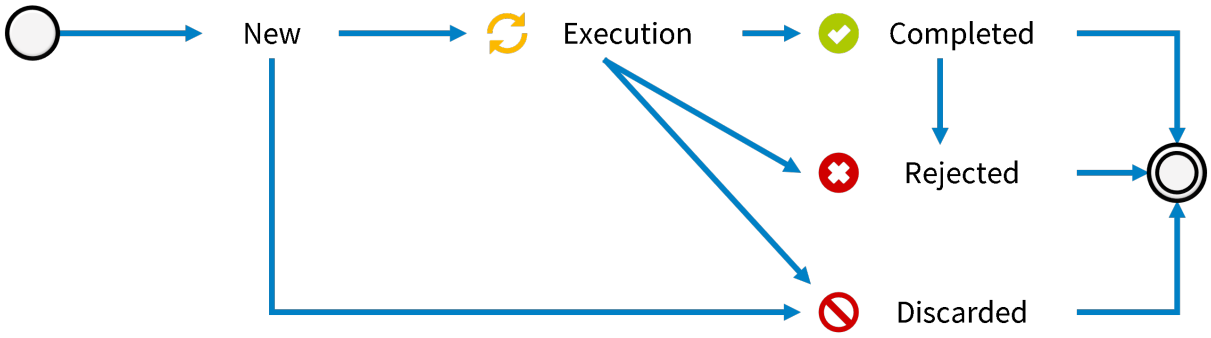


Fig. 3.3: The status network of a *task*

Status	Description
New	Initial status. The *task* is currently waiting to be run.
Execution	The *task* is currently being processed.
Completed	The *task* has been completed successfully.
Rejected	The *task* has been completed indicating a failure or rejection.
Discarded	The *task* has been cancelled without being processed.

Users can edit the following properties of *tasks* they are responsible for:

- *Responsible*,
- *Deadline*, and
- *Max. Duration*

This only applies to *tasks* in status **Execution**.

Note: These changes are logged in the activities of the *workflow* and responsible users will receive an email notification.

Users can also change the status of *tasks* in status **Execution** for which they are responsible.

Effects of a Task's Status Change

Changing the status of a *task* from **Execution** to another status will let the *workflow* continue with its next step. This will usually be the changing the status of subsequent *tasks* to **Execution**.

However, some status changes trigger additional effects:

- Changing the status of an approval *task* to **Rejected** leads to an immediate termination of the *workflow* with the status **Failed**.
- Changing the status of an examination or approval *task* to **Completed** with the option *Complete Prematurely* (page 9) also changes the status of its parent *task group* to **Completed**.

Another special case occurs when a *constraint* violation happens after changing the status of a *task* to **Execution**. In this case, the status of the *task* will be immediately changed to **Discarded**. Sequential successors

of the *task* are also set to **Discarded**, while parallel successors remain untouched. If the *constraint* violation cancels all remaining *tasks* of the *workflow*, the *workflow's* status will be set to **Failed**, otherwise the success of *workflow* depends on the remaining *tasks*.

3.1.4 Workflow Completion

Before a *workflow* is terminated with status **Completed** or **Failed**, it will run its *completion task* (if it exists). Only when the *completion task* has terminated, the *workflow* is set to its final status.

Within the *completion task*, the distinction if the *workflow* will fail or not can be made using the *constraint* "wf-designer: process completing successfully":

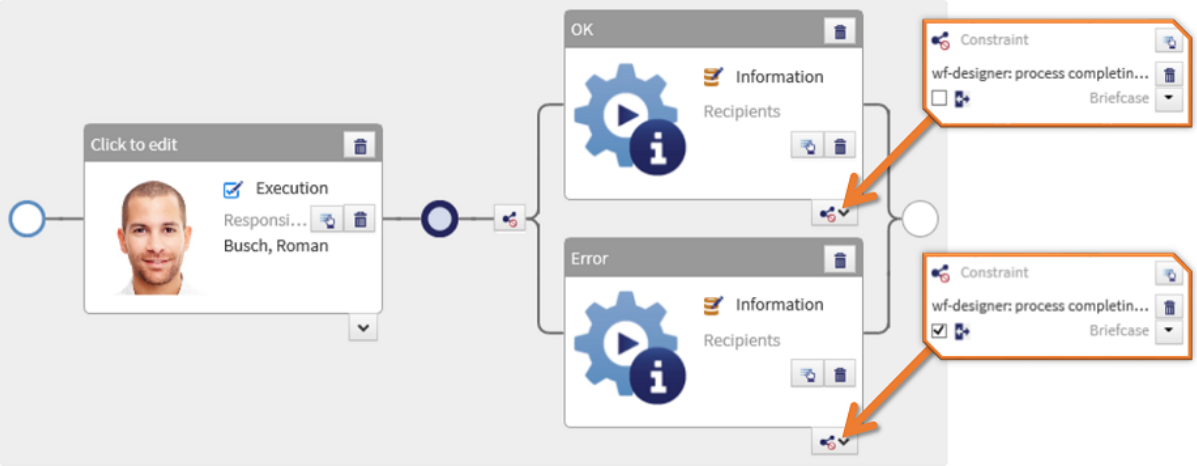


Fig. 3.4: Differentiation in the *completion task*

3.2 Processing Interactive Tasks

Users can search the system for *tasks* in status **Execution** they are responsible for and perform a status change as usual. However, *CONTACT Tasks* offers a dedicated interface and more streamlined support for processing *tasks*. Among other features, *CONTACT Tasks* offers

- a personal, configurable view of all current *tasks* (not limited to *workflow* tasks) and
- compact display of the relevant information in the detail area (context and description of the *task* as well as assigned *briefcase contents* and previous activities).

For more information please see the user manual of *CONTACT Tasks*.

CHAPTER 4

Monitoring Workflows

You can use a regular search to get an overview of your current *workflows*. To do this, double-click on the item *Processes* in the navigation pane.

In the search dialog, you can enter your user name in the field *Responsible* to find only *workflows* you are responsible for personally. Also, limiting the search to status  **Execution** is recommended.

The table you will be shown now contains your current *workflows* including their respective start and due dates.

Note: To get an overview of the current steps of your *workflows* at a glance, you can also search in *Processes* -> *Details* -> *Task* for *tasks* in status  **Execution**.

4.1 Protocol of a Workflow

The *protocol* of a *workflow* records the following events that occur when processing the *workflow* and its *tasks*:

- Status changes including any user comments,
- results of *constraint* checks,
- reservation of *tasks*,
- changes users made to *tasks* in any other status than **New**,
- unexpected events while processing *system tasks*,
- asynchronous continuation of *system tasks*, and
- errors leading to a premature termination of the *workflow*.

4.2 Email Notifications

The system offers users responsible for a *task* the option of being notified automatically by email when certain events occur.

4.2.1 Setup

Users can define themselves if they want to receive notifications by email in their personal settings. To do so, in the client's application menu open *Settings -> Personal settings* and switch to the tab *Notifications*.

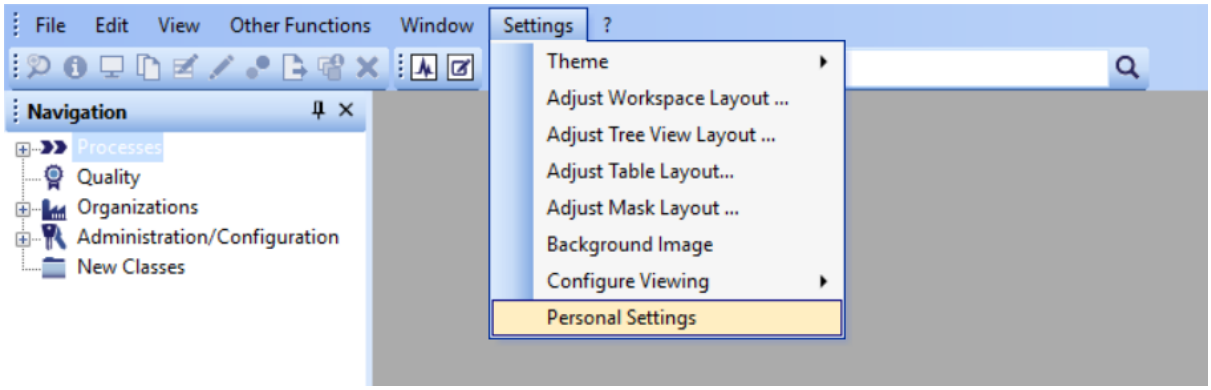


Fig. 4.1: Personal Settings

The option "When I Receive a New Task" activates or deactivates email notifications for this user.

Warning: If you do not receive any notifications despite having activated the setting, please check the email address of your user account.

If it is correct, please contact your system administrator.

4.2.2 Events with Notification

Email notifications are sent whenever one of the following events occur for a *task* in status **Execution**:

- the status of the *task* is set to **Execution**,
- objects are added to or removed from the *briefcase contents* of the *task*,
- a file is changed that is either itself contained in the *briefcase contents* of the *task* or whose business object is, or
- one of these attributes of the *task* is changed:
  - *Responsible*,
  - *Deadline* or
  - *Max. Duration*.

Note: A *workflow* itself does not trigger any email notifications. If you want to inform a certain group of people about the success or termination of a *workflow*, you have to include the notification explicitly in the structure of the *workflow*. You can use *system tasks* of type Information in the *completion task* as shown in *this example* (page 20).

The sender of an email notififcation is

- the user triggering an interactive event or
- one of the users responsible for the *workflow* (in case of an event occurring automatically).

Recipients are always the users responsible for the *task*, except for:

- The user triggering an event does not receive a notification themselves and

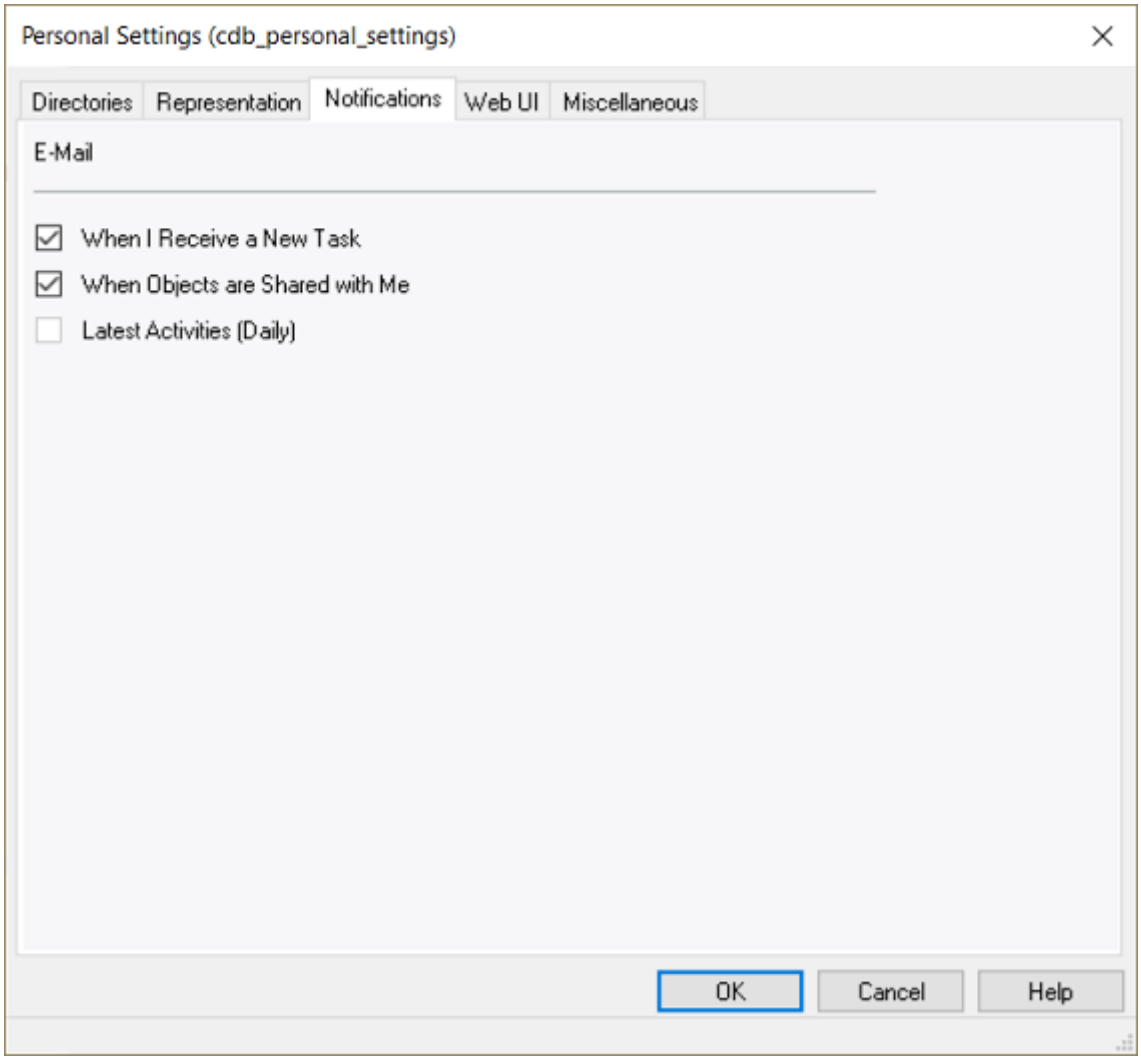


Fig. 4.2: Notification Settings

- notifications pertaining to reserved *tasks* are only sent to the user who reserved it.

CHAPTER 5

Permissions

A user's permissions on a *workflow* depend on

- ownership of the *workflow*,
- the *workflow's* status and
- role memberships of the user.

Permissions on a *task* additionally depend on its own status.

5.1 Workflow Owner

Every *workflow* has one or more responsible users, but only up to one *workflow owner*.

The *workflow owner* is automatically set when the status of a *workflow* changes to **Execution**. The *workflow owner* has a special meaning for the *workflow*:

- The *workflow* is run using the effective permissions of the *workflow owner* and
- *protocol* entries are written in the name of the *workflow owner*.

5.2 Effective Permissions

Most permissions in the context of *CONTACT Workflows* depend on permissions on the parent *workflow*.

Read permissions are not restricted for any objects used in *CONTACT Workflows*.

Protocol entries may be written (i.e. added) by everyone regardless of the status of a *workflow*.

The following abbreviations are used for permissions:

Permission	Description	Internal Name
Change	Summarizes write permissions. The object can be created, changed or deleted.	WRITE ACCESS, CHANGE
Change structure	Just like "Change", but referring to the structural children of the object (*task groups*, *tasks*, ...).	edit schema
Change status	The status of the object can be changed. May also be restricted to a specific target status.	process manage, task complete, task_ready, wf_form_templ_status

Note: In the following tables, permissionts granted to "Nobody" actually are explicitly denied to all users through exclusive grants.

Note: Permissions granted to "Everybody" in the following tables actually stands for members of the role `public`, excluding external users of the *CONTACT Collaboration Portal* especially.

5.2.1 Permissions on Workflow Templates

Condition	Permission	Granted for
None	• Change status	• cdbwf: Process Library Manager
Status **New**	• Change • Change structure	• cdbwf: Process Library Manager
Status *not* **New**	• Change • Change structure	• Nobody

5.2.2 Permissions on Workflows

Condition	Permission	Granted for
None	• Create • Change status	• cdbwf: Process Administrator
None	• Change status	• Responsible users
Status **New**	• Change • Change structure	• cdbwf: Process Administrator
Status **New**	• Change • Change structure	• Responsible users
Status **Execution**	• Change structure • Change status to – **Execution**, – **Completed** or – **Rejected**	• cdbwf: Process Administrator
Status **Execution**	• Change structure • Change status to – **Execution**, – **Completed** or – **Rejected**	• Responsible users
Status **Frozen**	• Change structure • Change status to – **Execution**, – **Completed** or – **Rejected**	• Nobody

5.2.3 Permissions on Tasks

Condition	Permission	Granted for
Status **Execution**	• Change • Reserve task • Remove reservation • Change status to – **Completed** or – **Rejected**	• Responsible users
Status **Completed**	• Delete • Change structure	• Nobody
• Status **Rejected** or • Status **Discarded**	• Change • Change structure	• Nobody
Workflow's status **Frozen**	• Change • Change status to – **Execution**, – **Completed** or – **Rejected**	• Nobody

5.2.4 Permissions on Form Templates

Condition	Permission	Granted for
Status **New**	• Everything (unrestricted)	• Administrator • Administrator: Master Data • cdbwf: Process Administrator • cdbwf: Process Library Manager
Status *not* **New**	• Read • Change status	• Administrator • Administrator: Master Data • cdbwf: Process Administrator • cdbwf: Process Library Manager

5.2.5 Permissions on Briefcase Contents

Please read the section Automatic Permissions.

CHAPTER 6

Examples

This chapter contains some examples of real-world requirements solvable with *CONTACT Workflows*.

Warning: Although the following section also contains notes on production use, productively used *workflows* should always be reviews extensively for correctness both from a technical and a business perspective.

Your system administrator can import the example *workflows* shown here using the provided script `example_workflows.py` as follows:

```
powerscript -m cs.workflow.updates.examples --autoinstall
```

6.1 Request for Quotation

Warning: This example requires *CONTACT Documents* (cs.documents) and utilizes *CONTACT Collaboration Portal* (cs.cp) for demonstration purposes.

This example primarily demonstrates the use of *forms* by means of a procurement process.

6.1.1 Usage

Create a document which describes the quotation request. Then run the operation `Workflow -> Create from Template` in the document's context menu. Select "Request for Quotation" and confirm the dialog.

Note: You can also describe the request directly in the *workflow*. To do so, run the operation `Create from Template ...` from the navigation pane's *Processes* entry. In the *Workflow Designer*, you can then change the description of the first two "Quotation" *tasks* accordingly.

The instantiated *workflow* will be opened in the *Workflow Designer*. Since all necessary parameters are already set, you may start the *workflow* immediately.

The *workflow* works as follows:

Fig. 6.1: Workflow Template *Request for Quotation*

- Two suppliers must each fill in a *form* with a quoted price and optional delivery date.
- Subsequently, a member of the role "Management: Procurement" has to select one of the two offers. To do so, one of the *forms* must be added to the *briefcase* "Selected Quotation" for the following steps and the *task* must be completed.
- If the selection has not been rejected, a member of the role "Procurement" now has to place an order with the selected supplier.

6.1.2 Special Aspects

Both suppliers have an "external" user account, e.g. these users can only log in using *CONTACT Collaboration Portal*.

Hint: If you do not have *CONTACT Collaboration Portal* installed or licensed, you can simply assign the role "public" to the users to simulate the *workflow*.

6.1.3 Notes on Usage

In productive use, an RFQ process requires selecting participating vendors. This can be modeled as an "execution" *task* in the first position. The responsible user can then create the *task* "Request for Quotation" for each supplier in a common parallel *task group*.

6.2 Release Part

Warning: This example requires *CONTACT Virtual Product* (cs.vp).

This example shows how a *workflow* can control the release process of a part.

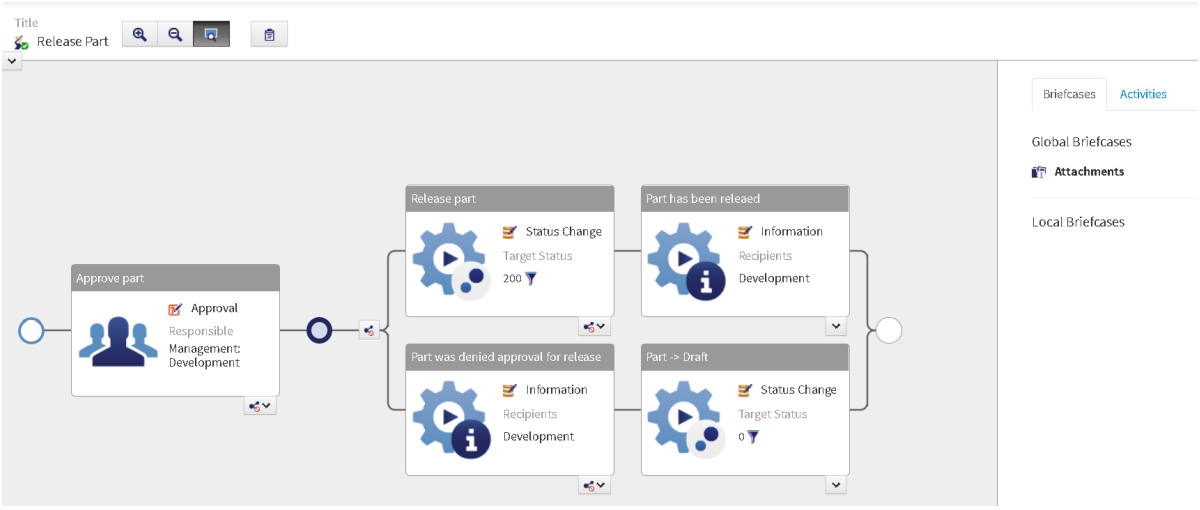


Fig. 6.2: Workflow Template *Release Part*

6.2.1 Usage

Run the operation `Workflow -> Create from Template` in the context menu of a part of status "Review". Select "Release Part" and confirm the dialog.

The instantiated *workflow* will be opened in the *Workflow Designer*. Since all necessary parameters are already set, you may start the *workflow* immediately.

The *workflow* works as follows:

- A member of the role "Management: Development" must first approve or deny the part's release.
- If the release is approved, the part's status is automatically set to "Released" and all members of the role "Development" are notified.
- However, if the approval is denied, all members of the role "Development" will be notified and the part's status will then be reset to "Draft".

6.2.2 Special Aspects

This *workflow* is tailored to parts in multiple aspects:

- The *template* can only be used with a part of status "Review" because it is assigned to an appropriately formulated object rule,
- the approval task contains a *constraint*, so that it is canceled immediately if the global briefcase "Attachments" does not contain exactly one part of status "Review", and
- the status changes contain filter rules that also only apply to part of status "Review". All other *briefcase contents* are ignored by the status changes.

In the *completion task*, the upper *tasks* handle the case that the *workflow* succeeds. The lower ones handle the *workflow's* failure.

The positioning of the information tasks anticipates possible errors when changing the status:

- The information about the release only becomes active after it has actually happened, but
- the information about the denied approval for release becomes active before the part's status is reset to "Draft".

6.2.3 Notes on Usage

The following changes are recommended for actual use:

- The *workflow* should be instantiated and started automatically after changing a part's status to "Review".
- Responsible users or roles should be dynamically determined for more precision. In particular, the information about a denied approval for release should only go to the user triggering part's status change to "Review".

6.3 Engineering Change Order

Examples concerning Engineering Changes can be found in the application package *CONTACT Engineering Changes* (cs.ec).

CHAPTER 7

Glossary

Completion Task

Special *task group* which is executed after the regular end of its *workflow*. The term is used synonymously in the *Workflow Designer* with *tasks* within this *task group*.

Completion tasks are usually used for exception handling or notifications.

- Example

Task

Element of a *workflow* that combines input data with a task description. The description is textual for interactive tasks and a PowerScript function for *system tasks*. Input data is linked to the task via *briefcases*. The behavior of a :term‘task‘ is determined by its Task Types.

- Creating Tasks
- Status Network of a Workflow Task

CONTACT Tasks offers users an overview of their tasks in status **Execution** and the opportunity to easily complete them.

Edit Mode

Controls the Accessing Folder Contents. Possible values:

Database Value	Name	Permissions Granted (Example)
0	**Info**	Read (read, read_file)
1	**Edit**	Write (accept, lock, save, unlock)

- Details
- Setup

Constraint

You can use constraints to define conditions that are checked immediately after the status change of a *task* in **Execution**. If there is at least one violated constraint for the *task*, the status of the *task* is set to **Discarded**. Constraints are object rules.

- Usage
- Setup

Form

Forms enable capturing and processing structured data (metadata) in *workflows* and their *tasks*.

A form is always instantiated from a *template*.

- Usage
- Setup

Briefcase

Briefcases aggregate input or output data (see *briefcase contents*), either for an entire *workflow* ("global briefcase") or individual *tasks* ("local briefcase").

The assignment of a briefcase to a *workflow* or a *task* additionally contains an *edit mode* for additional access rights management.

- Usage
- Setup

Briefcase Contents Object assigned as content to a *briefcase*. Possible contents are *forms*, files, or business objects that serve as input or output data of a *workflow* or a *task*.

Parameter

Filter Parameter *System tasks* may contain parameters serving as input for processing them. Parameters of the *system task definition* are mandatory.

Protocol The protocol of a *workflow* contains messages (both regular and unexpected events) and their timestamps that are useful for monitoring and subsequent analysis.

- Additional Information

Task Group

Task Groups are containers for *tasks*, which do not have their own logic apart from the execution order of their child *tasks*.

Sequential task groups run their child *tasks* strictly one after another, parallel task groups run them at the same time. Task Groups only reach status **Completed** when all child *tasks* are in a final status (**Completed**, **Rejected** or **Discarded**).

Workflows behave like sequential task groups themselves.

Task groups are not explicitly displayed in the *Workflow Designer*. *Tasks* displayed in a column share a parallel parent task group, *tasks* displayed in a row are either direct children of the *workflow* or of a mutual sequential parent task group.

System Task

A *task* which is not processed interactively by a user, but automatically by a PowerScript function. System tasks in status **Execution** are run asynchronously by the *workflow service*.

- Usage
- Setup

System Task Definition Logic, name and appearance of a *system task* are determined by their definition. Each *system task* must reference a definition via its attribute `task_definition_id`.

Your system administrator can add new definitions to expand the stock of usable *system task* types. Please read the section in the administrator's manual.

Template A template is an object that cannot be used itself, but can be copied to get a usable object. Copying a template is called "instantiating".

Objects that can be created from templates are *workflows* and *forms*.

Workflow

A workflow contains temporal and logical dependency structures between *task groups* and interactive and automatic *tasks*. It is able to model entire business processes in some instances.

A workflow may also be a *template*.

- Menu access: *Processes –> Workflows*

Workflow Designer

Web application supporting the graphical design of a *workflow*.

- Usage

Workflow Owner The user who changes the status of a *workflow* to **Execution** will from then on be considered its "owner". They are personally responsible for the correctness of the *workflow* design at that time.

Workflow Service This service runs *system tasks* in status **Execution**. It is implemented as a "Message Queue". Whenever a *system task's* status is set to **Execution**, a job which controls the execution by the service is created.

The service's name is *cs.workflow.services.WFServer*, that of the queue *wfqueue*.

- Additional Information
- Message Queues
- Services

Workflow Category Categories allow for logically grouping *workflows*. They let users preselect *workflow* templates more easily. Possible category names could be *Tender*, *Engineering Change Order* or *Request for Information*.

- Menu access *Administration/Configuration -> Catalog Administration -> Processes -> Workflow Categories*

List of Figures

2.1 The *Workflow Designer* . . . 3
2.2 The header of the *Workflow Designer* . . . 4
2.3 Drawing a rectangle around one or more existing *tasks* . . . 10
2.4 Buttons for creating a *task* at each possible location . . . 11
2.5 Successor after clicking the right plus button . . . 11
2.6 Parallel *task* after clicking the bottom plus button . . . 12
2.7 Adding a *form* to a *task* . . . 14
2.8 Opening a *task's* the *forms* . . . 14
2.9 Example concatenation of *form*, permission and *system task* "Run Operation". . . . 15

3.1 The status network of a *workflow* template . . . 17
3.2 The status network of a *workflow* . . . 18
3.3 The status network of a *task* . . . 19
3.4 Differentiation in the *completion task* . . . 20

4.1 Personal Settings . . . 22
4.2 Notification Settings . . . 23

6.1 Workflow Template *Request for Quotation* . . . 30
6.2 Workflow Template *Release Part* . . . 31

List of Tables

Index

B

Briefcase, **34**
Briefcase Contents, **34**

C

Completion Task, **33**
Constraint, **34**

E

Edit Mode, **33**

F

Filter Parameter, **34**
Form, **34**

P

Parameter, **34**
Protocol, **34**

S

System Task, **35**
System Task Definition, **35**

T

Task, **33**
Task Group, **34**
Template, **35**

W

Workflow, **35**
Workflow Category, **35**
Workflow Designer, **35**
Workflow Owner, **35**
Workflow Service, **35**