Signavio Workflow Accelerator
User Guide

3.72

Contents

Skip to the *Tutorials* (page 168) if you don't want to read a manual.

Chapter 1

License types

To log in to Signavio Workflow Accelerator, your user account must have a user license that has not expired. You can choose between several license types:

- Enterprise Cloud Free Trial
- Enterprise Cloud
- Academic Cloud

The *Feature Overview* section on the Workflow Accelerator product page[1] shows the main differences between these versions. These differences affect:

- which software features you can use
- the amount of user file storage
- the support level.

The *Billing* (page 117) page in the application shows details of the current license type.

1.1 Enterprise Cloud Free Trial

The free trial license lets you try the full version of the software for a limited period without purchasing a license. To create a trial license, use the registration page (Europe server[2], US server[3], Australia server[4]). When the trial expires, Workflow Accelerator will send you an email inviting you to purchase licenses.

1.2 Enterprise Cloud

A *Enterprise Cloud* license gives you full access to all features with minimum limitations. To purchase *Enterprise Cloud* licenses, contact sales@signavio.com[5].

1.3 Academic Cloud

A free *Academic Cloud* license allows university students to use Workflow Accelerator to learn about workflow management. You may only use this license if you have a valid university email address.

[1] http://www.signavio.com/products/workflow/
[2] https://workflow.signavio.com/registration
[3] https://workflow-us.signavio.com/registration
[4] https://workflow-au.signavio.com/registration
[5] sales@signavio.com?subject=Signavio%20Workflow%20Enterprise%20Cloud

You may only use this license for educational purposes and not for administrative processes in an educational institution.

To register for an academic license, first register for the free trial. Then use the *academic version* link at the bottom of the purchasing page (Europe server[6], US server[7]) to extend the license period for one year. Alternatively, use the *Student? Click here* link at the bottom of the *Billing* (page 117) page in the application.

[6] https://workflow.signavio.com/buy
[7] https://workflow-us.signavio.com/buy

Chapter 2

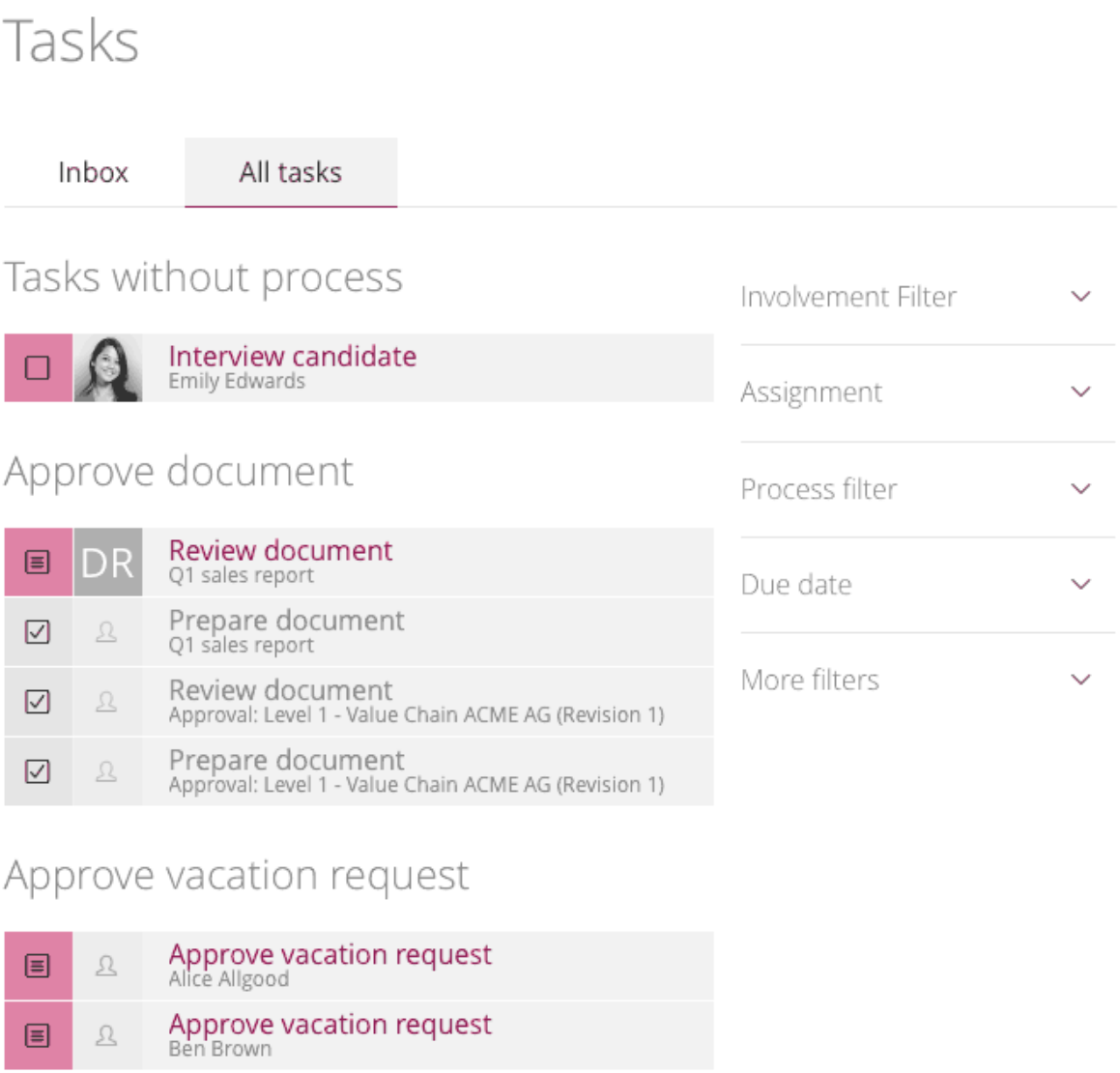
Tasks

A task represents work that someone will presumably complete. *Cases* (page 13) typically include multiple tasks, usually those that the process defines. You can also add ad-hoc tasks to a case.

In Signavio Workflow Accelerator, you can assign a task to a specific user, set a due date and add subtasks.

2.1 Viewing all tasks

The *All tasks* view shows tasks for all assignees.



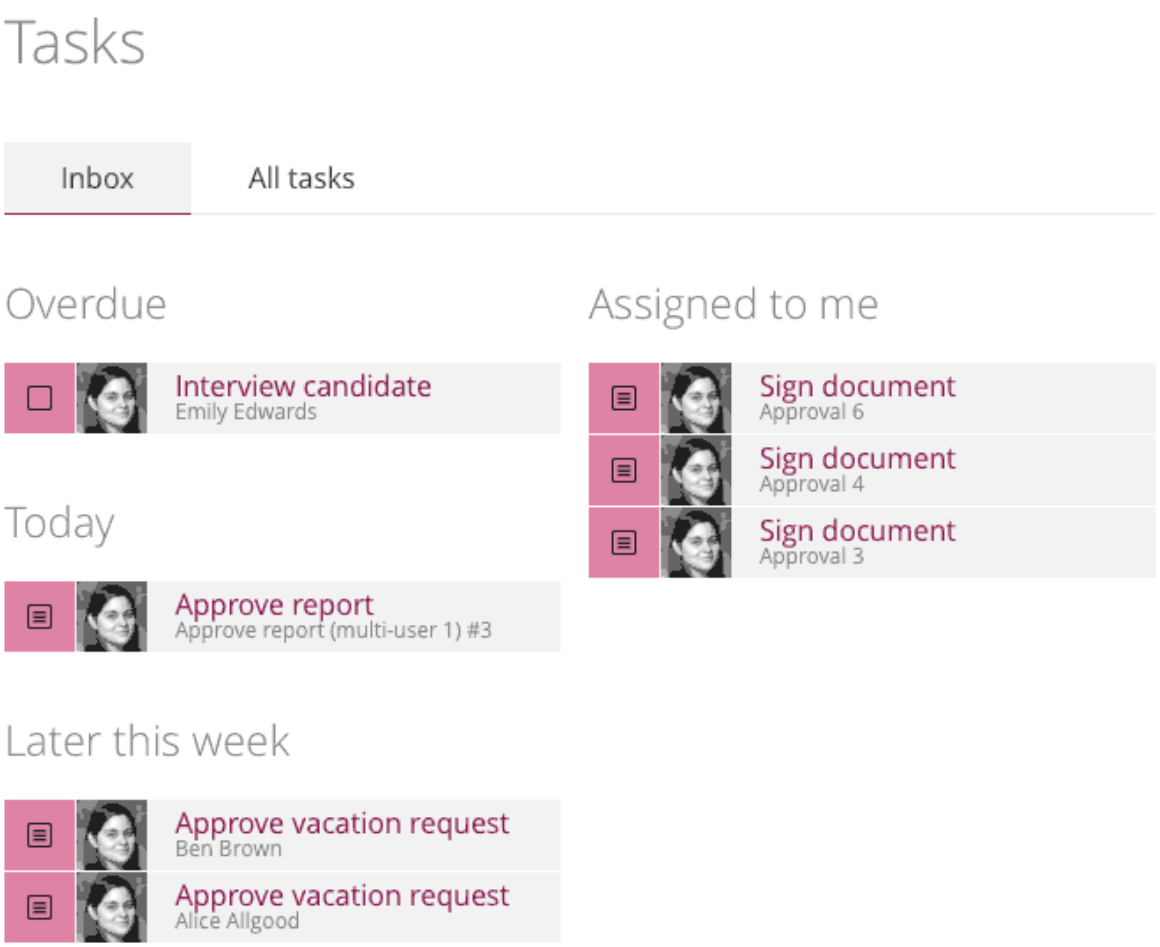
The list of all tasks

This view includes filters that you can use to sort through a large list of tasks.

- *Involvement filter* - shows tasks according to how they relate to you, such as tasks you started.
- *Process* filter - shows tasks for a specific process.
- *Due date* filter - shows tasks according to their due dates, such as only overdue tasks.
- *Completed* filter - shows complete tasks, which the task list normally excludes.
- *Assignee filter* - shows tasks that have a specific assignee, or tasks that have a specific candidate.

2.2 Viewing your tasks in the Inbox

The *Inbox* shows an overview of your assigned tasks: a list of tasks for you to work on. To open the Inbox, select *Tasks* from the main menu and select the *Inbox* tab.



The tasks Inbox

Each task shows the assignee - you for all Inbox tasks - and a link to the task itself.

The left-hand side of the inbox has sections for tasks that have due dates, so you can prioritize your work. The right-hand side lists tasks that don't have a due date.

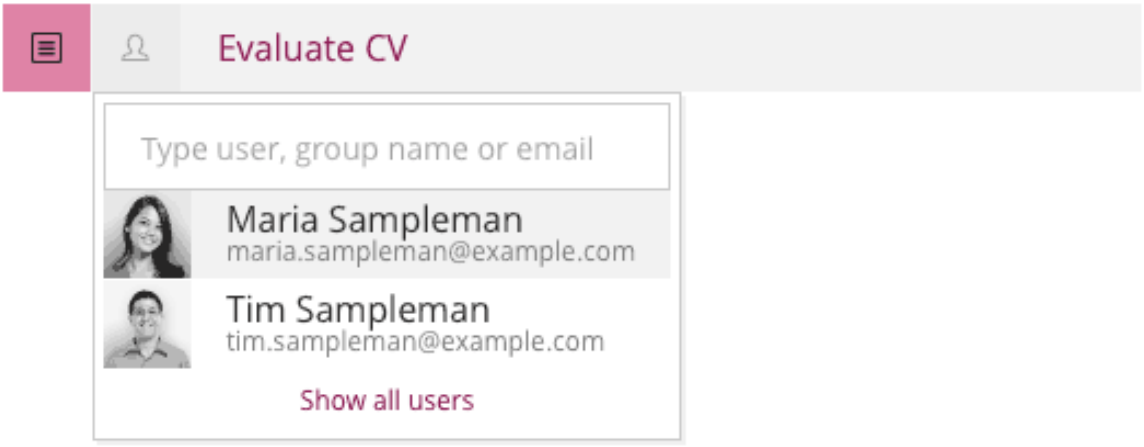
2.3 Creating case tasks

To create tasks, open a case, and use the tasks list to add a new task. Enter the task title in the text box to create the task. After you have added a task, it appears in the list. When a process starts, Signavio Workflow Accelerator creates a *case* (page 13) and starts all elements that do not have incoming transitions.

2.4 Assigning tasks

You can assign a task to yourself, or someone else, to indicate who you expect to work on the task. The assignee's *tasks inbox* (page 9) lists assigned tasks, and the assignee receives task *notifications* (page 120) and *reminders* (page 57).

You can assign a task from anywhere a task appears in a task list. First, click the assignee button immediately to the left of the task name, to open the list of candidates.



Assigning a task

To assign the task, select a candidate from the list, or type a name or email address to filter it first.

2.5 Completing tasks

There are two ways to complete a task. You can use the *Done* button on the task details page, or you can go back to the list of tasks and click the purple check button, as long as the task does not include a form.

When tasks have a form, the task details page displays the form and includes button(s) to complete the task at the bottom of the form. Normally there is only a single *Done* button, but the page will display multiple buttons when you use a decision. Each of those buttons will register the decision and complete the task in one go.

2.6 Task details view

Selecting a task opens the task details view.

When you open the *Task details view*, you will see three sections. The left-hand panel shows the lists of tasks that are associated with a particular case. The center panel shows all the details of a selected task. The right-hand panel shows comments and the history of a task, along with core information about your case. (You can also hide the right-hand panel, if desired.)

You can add ad-hoc subtasks using this view. Ad-hoc subtasks are tasks that are not included in the main process, but are added later as needed. The number of subtasks in a case is listed in the left-hand panel. To see details about a subtask, select a task.

2.7 Reopening tasks

When the task page shows a *Reopen* button, it means that someone has completed the task, but you can reopen it. You cannot reopen a task that has a form, because completing the task finalizes the form data. You may want to reopen a task that you closed by accident, or when you learn about additional work that belongs to this task. If you just forgot to attach a document or make a comment, you can still do that without reopening the task.

Reopening a task instead of creating a new task has the advantage that the existing task retains its context.

Chapter 3

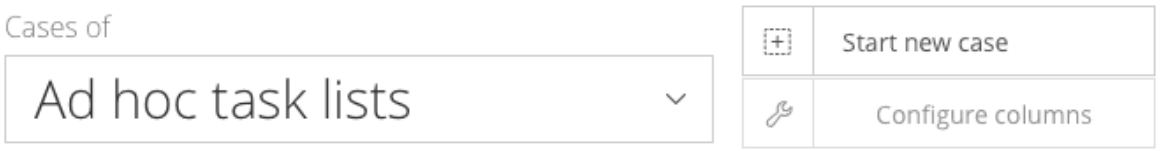
Cases

You can use a case as a small collaboration space for a particular goal (like “Hire employee”, or “Sign contract”, for example). Cases typically represent more work than a simple task for a single person, but less than a whole project. A case breaks the goal down into concrete action items (or tasks) so you can collaborate with other people. The case brings together a set of tasks, a discussion and documents, and allows participants to share any relevant context information for the tasks.

3.1 Starting an ad-hoc case

Signavio Workflow Accelerator supports two types of cases: cases that relate to a process and ad-hoc cases. An *ad-hoc case* does not have a predefined process. It creates a collaboration space that you can use to reach a one-off goal.

To create a new ad-hoc case, navigate to *Cases*; on the *Cases of* menu, select *Cases without a process*, then click *Start new case*.

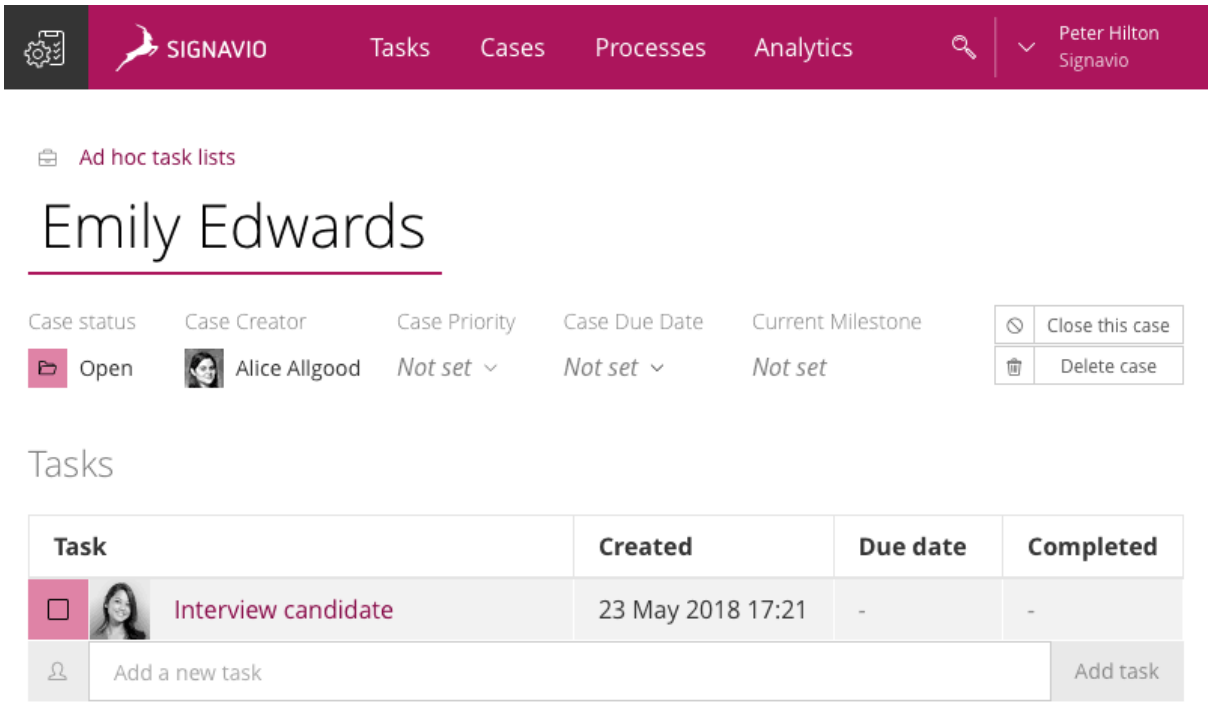


Start a new case

Then the case header appears.

Enter case name

Next, type the case title and hit Enter. Workflow Accelerator now creates the new case.



New case

3.2 Starting a new process case

A *process case* uses the latest version of the workflow defined by a published process. It creates a collaboration space for working towards a predefined goal.

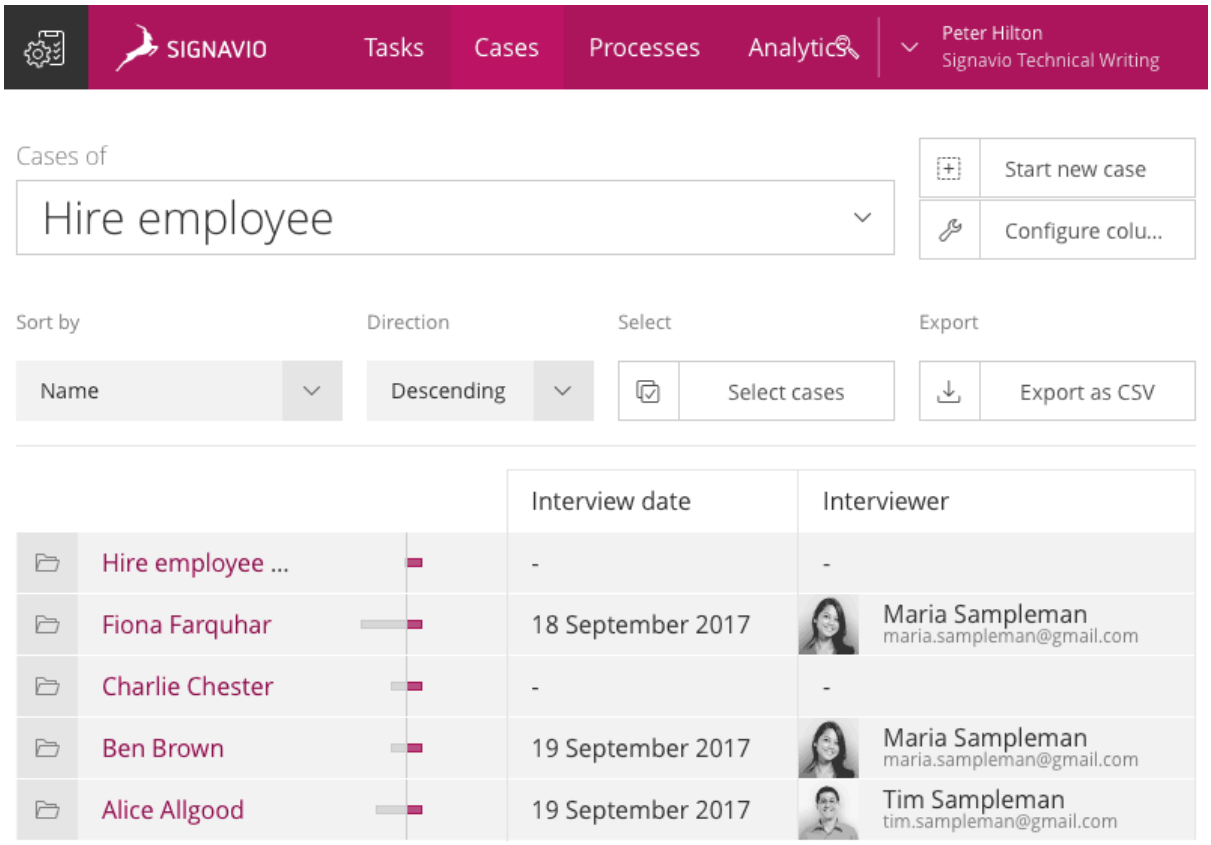
You can create a new process case by selecting the *Start new case* button in one of three places:

1. In the process builder, on the *Versions* (page 40) page, next to the latest published version
2. On the *Processes* page, next to the name of each published process
3. On the *cases overview* (page 14) page, for the selected process.

If the process does not define a trigger, then you enter a case name as when *Starting an ad-hoc case* (page 13). Other trigger types generate their own case names, or use a *case name template* (page 39).

3.3 Browsing cases

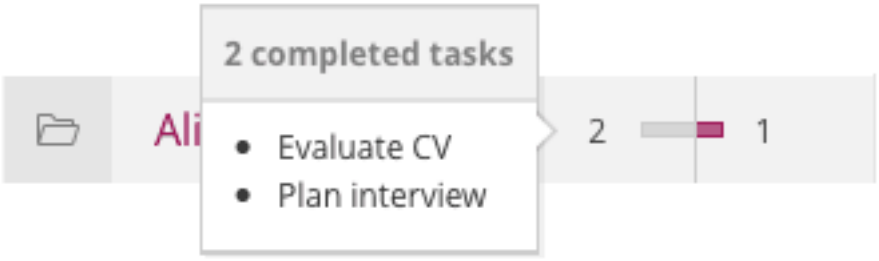
The *Cases* view shows an overview of cases for a single process. To open the Cases view, select *Cases* from the main menu.



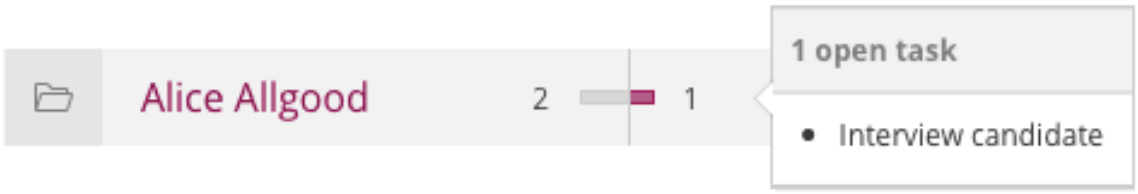
The Cases view - cases of the "Hire employee" process

The table shows cases for the *Hire employee* process. You can use the menu above the table to select a different process, ad hoc cases that don't have a process, or cases of a deleted process. Each case in the table show the case name, grey and purple task status bars, and additional columns that show the values of workflow *variables* (page 43).

The grey and green status bars show information about completed and open tasks when you hover the mouse cursor over them.



Hover over the grey bar to see a case's completed tasks.



Hover over the open bar to see a case's open tasks.

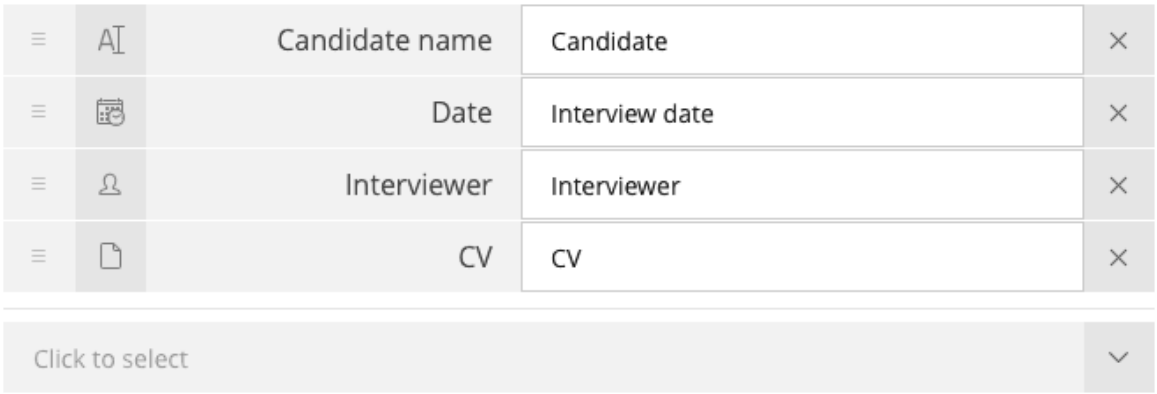
These status bars appear in the upper right of the workflow data columns, which you can customize.

Use the linked case name to open the case's details page.

3.4 Configuring case view table columns

The *Cases* view's table includes columns for workflow *variables* (page 43), which usually correspond to form fields on a trigger form or in a user task. You can select which fields the Cases view shows as table columns, so you can have a clear overview of the process' cases.

To customize the table columns, open the Cases view and click the top-right *Configure columns* button.



Configuring table columns

Use the drag icon on the far left of the list of columns to change the column order, and click the delete icon on the far right to remove a column. You can also use the text box to edit the column's heading.

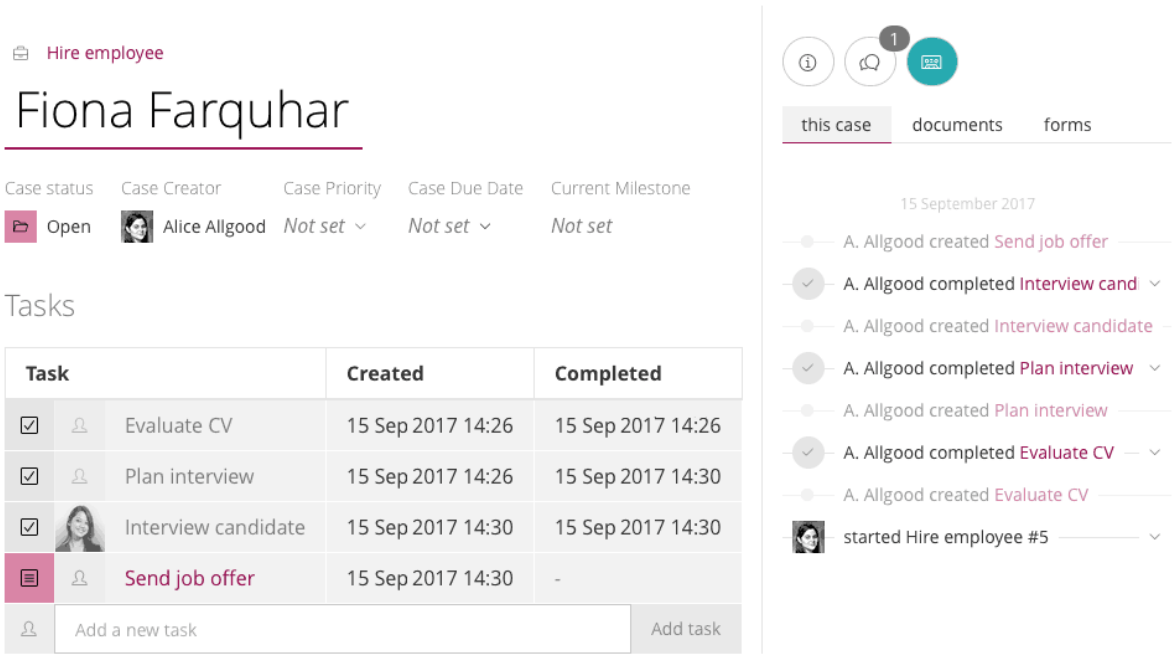
Below the list of columns, you'll find a menu for selecting additional column to add to the table.

Adding a column to the table

The menu lists all of the workflow variables that the table does not currently include. In this example, the *Interviewer* variable has the type *User*, which means that you can access additional fields for the user's email address, first name, ID and last name.

3.5 Viewing case details

The *Case details view* displays all tasks in a table, along with important information like **creation date**, **closing date**, and **due date**. It also shows the current milestones of a case. You can change the priority and due dates for a case in this view. The name of the process the case belongs to is displayed above the case name.

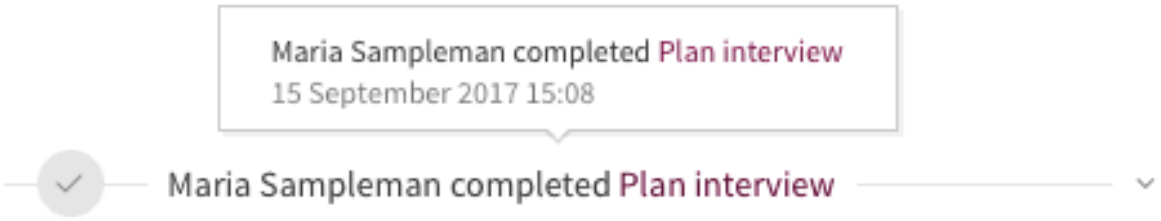


Case details view, showing three closed tasks, one open task and the info panel

Each case has an info panel, which itself is divided into three panels: core information, comments, and history. Whenever you open one of these panels, your selection is stored in your user preferences. The

next time you open the case details view, the panel you selected in your previous session will open by default.

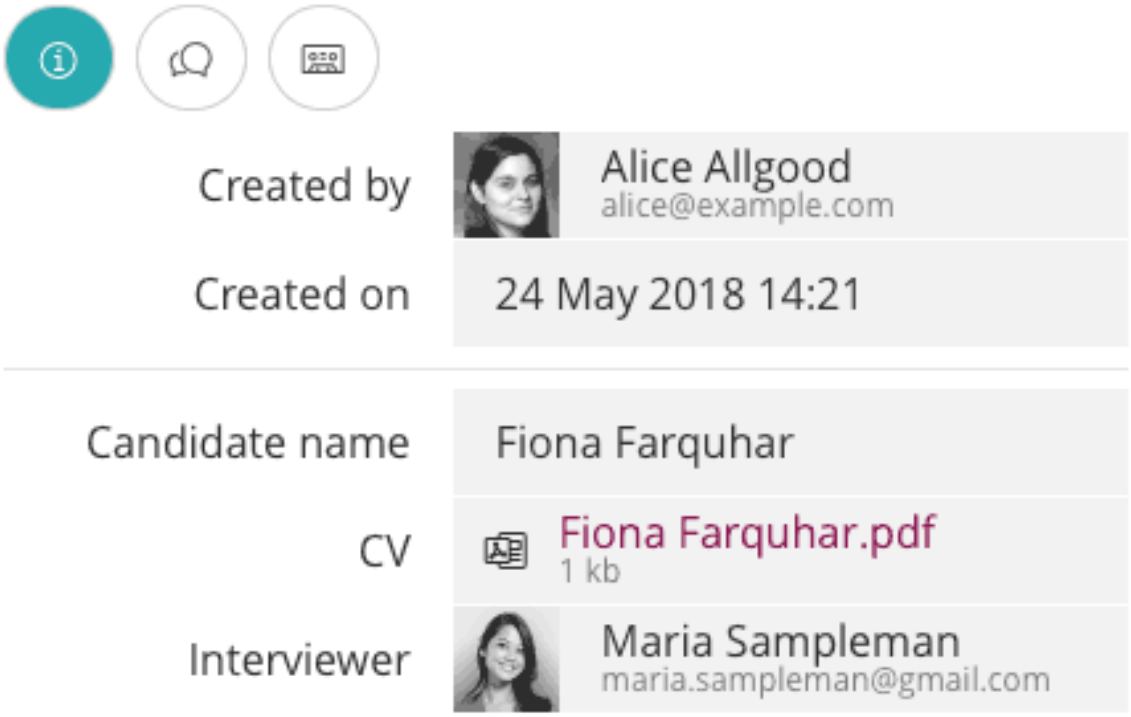
Hover over a history panel entry to open a pop-up that shows its exact date and time.



Hovering over an event stream entry to show its timestamp

3.6 Viewing core information

The core information feature allows you to see all important information about a case with a single glance.



The case details view's core information tab

You can find the core information in three locations: in the right side panel of the new case view; in the details tab of the workflow editor; and in the 'Set core information' action in the new process view. Core information can be edited or filled in with existing information from the workflow.

3.7 Viewing case history

The case history panel shows a timeline of events in a selected case. You can limit the information shown in the case history panel to just uploaded documents or submitted forms. To see the timestamp of an event, hover your cursor over it in the timeline.

When looking at a task, the case history panel will by default show only events related to the selected task. You can reset this by simply clicking the *this case* tab.

Note that by default, events in the history panel that include forms are collapsed by default. To view the form data, just click on the event.

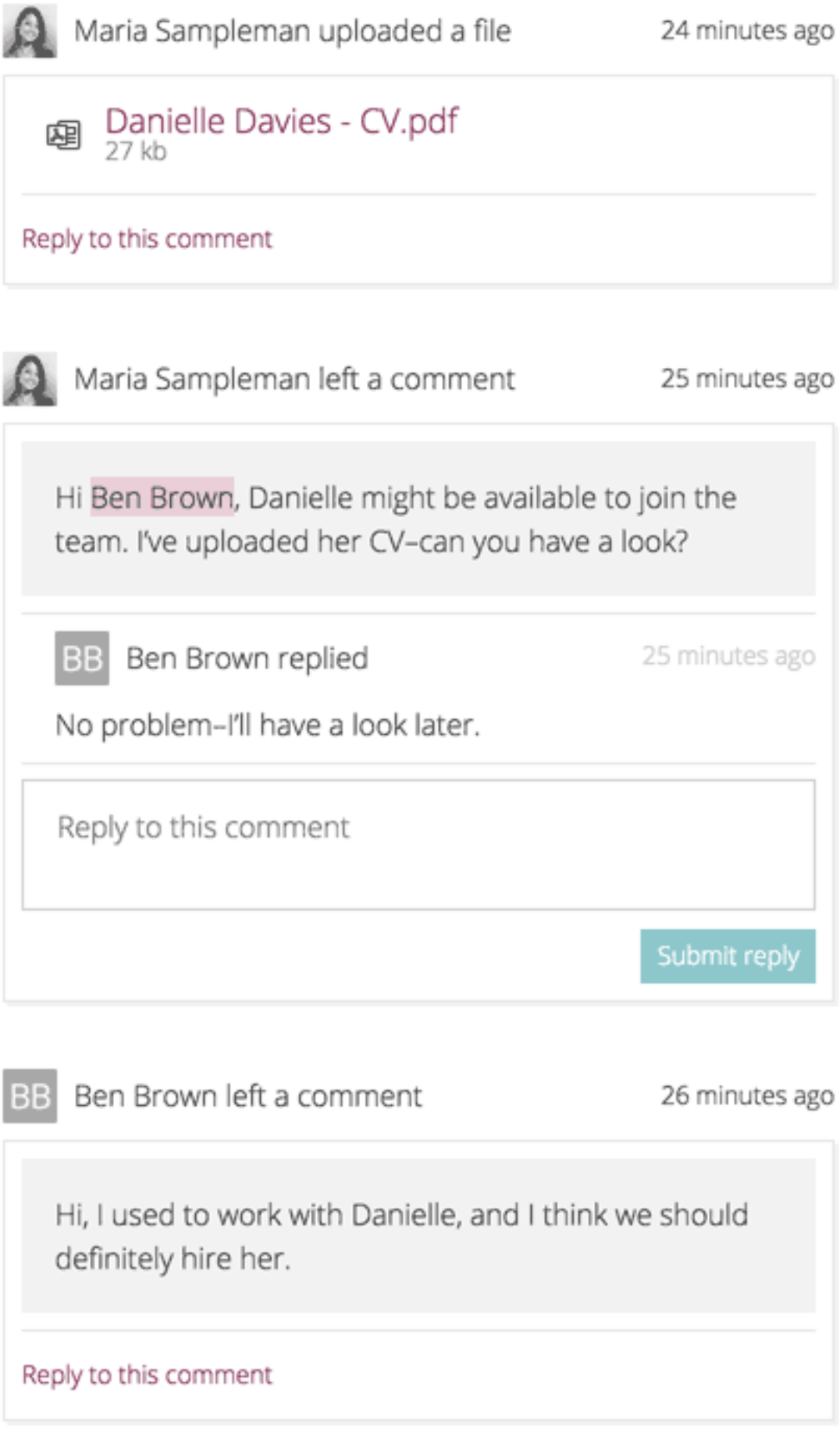
3.8 Other open activities

The case view contains the *Task* list. This list shows all open and closed tasks in a case. Subprocesses are listed under the *Subprocess* heading.

In most cases, completing the last task will close the case. However, sometimes a case must wait for a sub-process, or some other activity that doesn't correspond to a task in the *Tasks* list, to complete before closing. In these situations, you can close the open activity manually. These activities are usually timers, such as an *Intermediate timer event* (page 105). They can be found under the *Other open activities* heading, along with any execution errors that occurred during the process. To skip a timer, click the Skip button located next to the name of the timer.

3.9 Commenting on a case

Working on a case often includes collaboration with other people. You can use the case view to discuss the case with other people, which results in a discussion in the comment panel.


Maria Sampleman uploaded a file
24 minutes ago
Danielle Davies - CV.pdf
27 kb
Reply to this comment
Maria Sampleman left a comment
25 minutes ago
Hi Ben Brown, Danielle might be available to join the team. I've uploaded her CV–can you have a look?
BB
Ben Brown replied
25 minutes ago
No problem–I'll have a look later.
Reply to this comment
Submit reply
BB
Ben Brown left a comment
26 minutes ago
Hi, I used to work with Danielle, and I think we should definitely hire her.
Reply to this comment

By using case comments instead of email, participants ensure that discussions retain the full context for everyone involved.

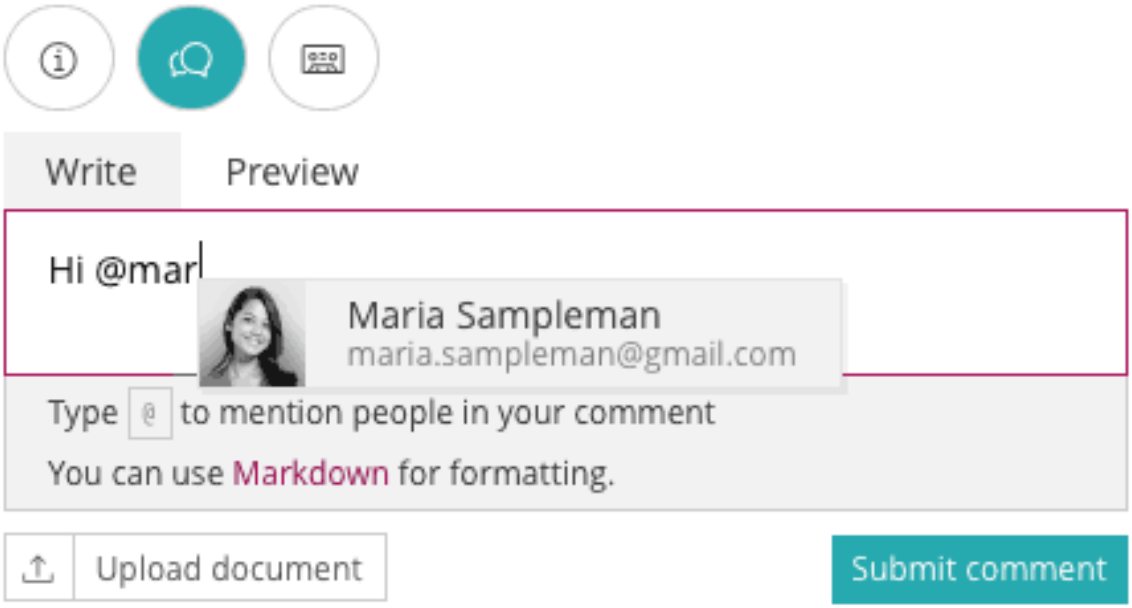
> **Note:** Once created, comments cannot be edited or deleted.

In the comment panel, reply to existing comments to keep the discussion structured. Hover over a user's avatar to see their full name. To see the time stamp of a comment, hover over the relative time. To add a comment, type in the text box above the comment panel.

Everyone who has access to the case can follow the discussion. Keeping the discussion 'inside' the case preserves the context, which makes it easier to follow than an email conversation.

You can add comments to specific tasks or cases. Comments that are assigned to a specific task can only been seen by people with access to that task. When a task is opened, the comment list is limited to task-specific comments by default. You can choose whether to comment on a specific task or on the entire case by selecting the corresponding tab beneath the comment box.

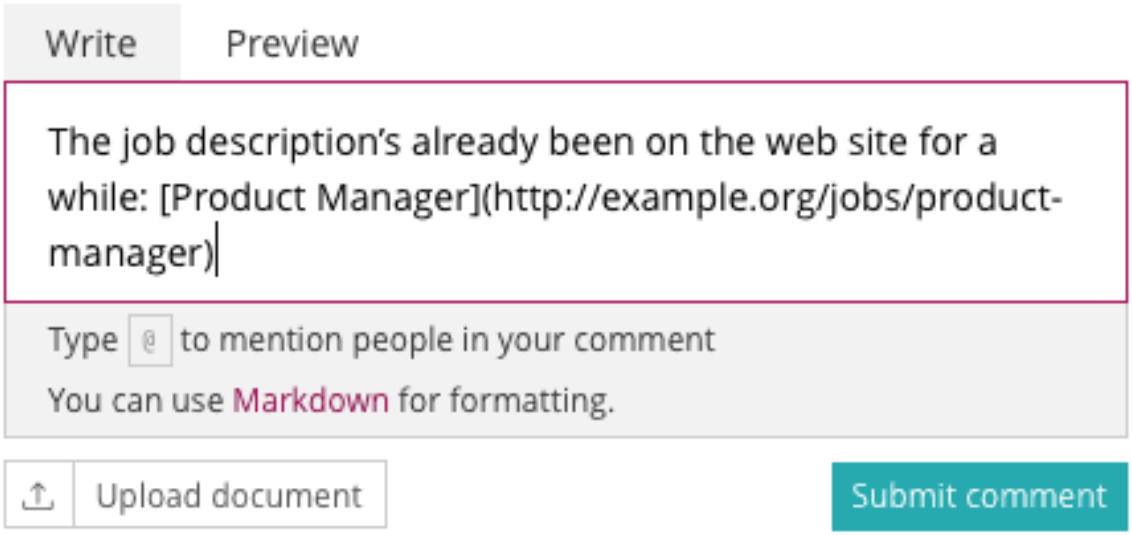
Sometimes, you want to direct a comment to a specific person. While entering a comment, you can 'mention' someone by typing a `@` and choosing their name from the list. When you mention someone in a comment, Workflow Accelerator sends them an email notification to bring them into the discussion.



Mentioning someone in a comment

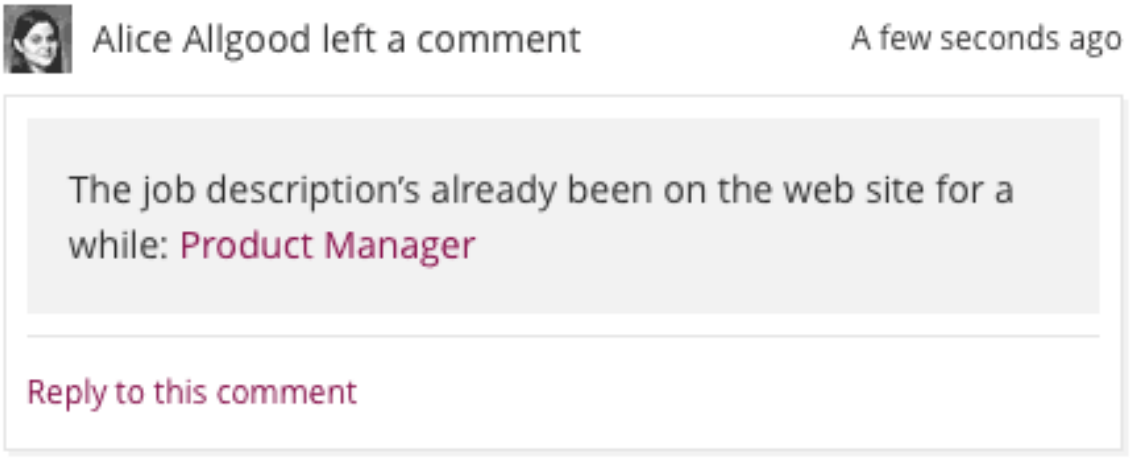
You can also mention two specific groups, corresponding to people working on the case. Enter `@all` to mention all participants in the case. Enter `@open` to mention the assignees of open tasks within the case.

You can also use *Markdown* (page 129) formatting in comments, for things like text styles, headings and lists.



Using Markdown formatting in a case comment

You may find Markdown most useful for adding links to external information that relates to the case.

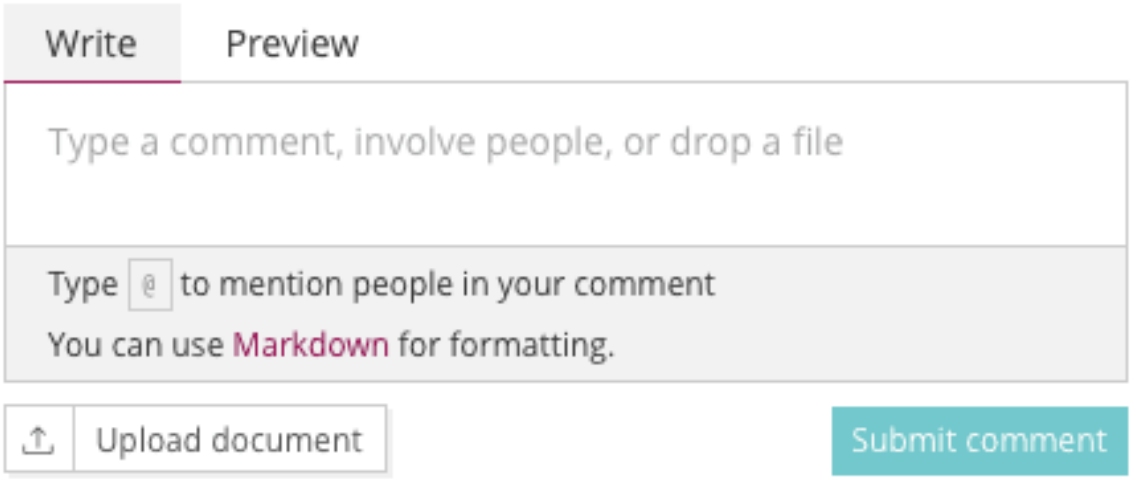


A hyperlink in a case comment

3.10 Attaching documents to a case

As well as commenting on a case, you can attach documents to share with the case's participants. For example, cases in a job vacancy process might require candidate CVs. Anyone with access to the case can download a case's documents, which makes them more useful than email attachments.

To attach a document to a case, select the the *Upload a document* option next to where you add comments.



A document attached to a case

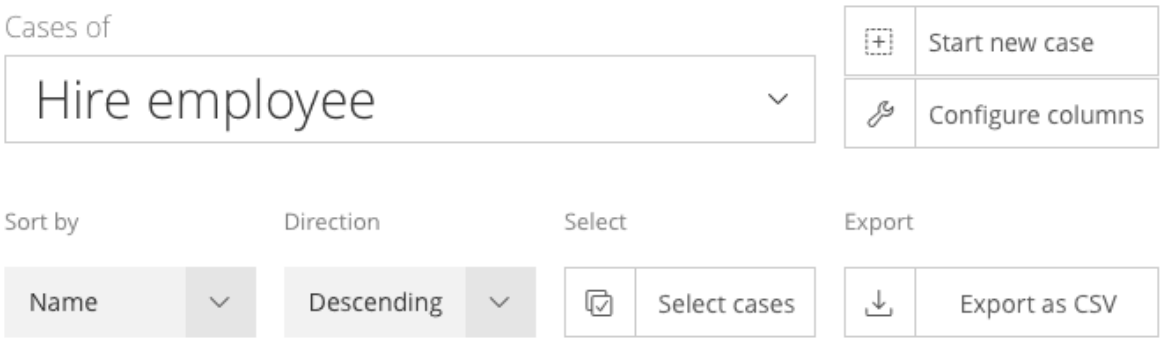
The uploaded document will be visible in the comment panel as well as the history panel.

If cases regularly require the same document as part of the process, you can make this clearer to people who work on cases by adding a file upload form field to a *user task form* (page 84).

3.11 Closing a case manually

You normally close a case by completing all of its tasks. However, sometimes you want to abandon a case and stop working on it. To do this you can manually close a case.

To close a case, simply click “Select cases” in the menu under the case name:

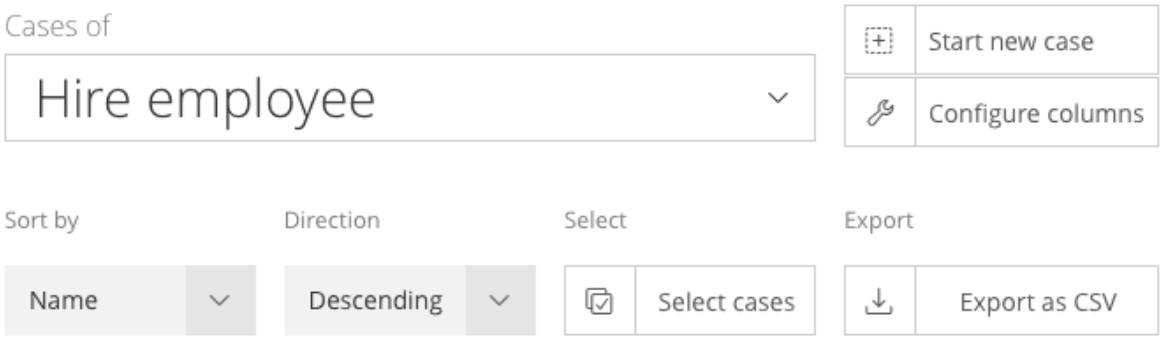


Click the check box next to the case you want to close. (You can select multiple cases at once.) Then click “Close all selected cases”.

3.12 Deleting cases

You do not normally delete cases in Workflow Accelerator: you close cases that you have finished working on. However, you sometimes do need to delete cases, such as the test cases that you create while developing the initial versions of a process model.

To delete a case, click “Select cases” in the menu under the case name.



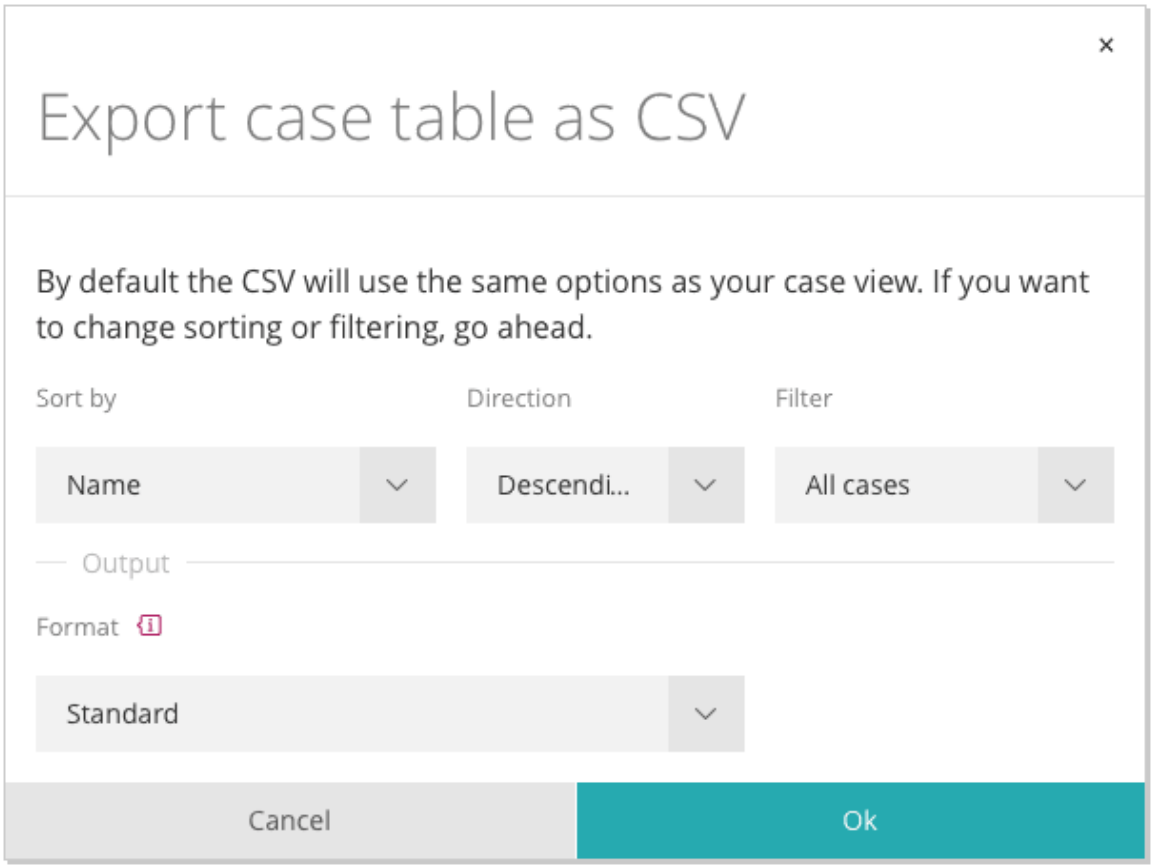
Use the checkboxes in the cases table’s first column to select which case(s) you’d like to delete. Then click “Delete all selected cases”.

You can also delete cases via the **Delete case** button in the case view.

3.13 Exporting cases data

You can export the information about a process’ cases to a CSV file[8] that you can open in a spreadsheet. To export case data, open the *Cases overview* (page 14), select a process, and then select *Export as CSV*. You may find this useful for reporting or auditing, for example.

[8] https://en.wikipedia.org/wiki/Comma-separated_values



CSV export options

Use the CSV export options to specify the ordering of exported cases, whether to filter by status (open or closed), and the output format. Try a different output format option if you have problems loading the exported CSV file into another application, such as Microsoft Excel. The output format options determine which characters the CSV output uses to quote and separate characters and lines:

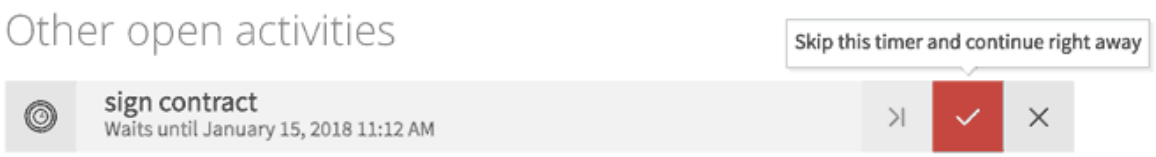
- *Standard* - conventional CSV format
- *Excel* - Microsoft Excel compatibility mode
- *Excel (Northern Europe)* - better Excel compatibility for some European countries
- *Tabs* - separate values with tabs instead of commas.

The CSV export uses *UTF-8 text encoding*. Select *UTF-8* when opening the CSV in Microsoft Excel, for example, to preserve characters such as letters with accents.

3.14 Skipping intermediate timer events

When a process includes an *Intermediate timer event* (page 105), case execution waits for the timer to complete before continuing. Sometimes, you want to continue immediately without waiting for the timer.

You can manually skip a timer, without waiting for its configured delay. The *case details* (page 17) view shows open timers under the *Other open activities* (page 19) heading.

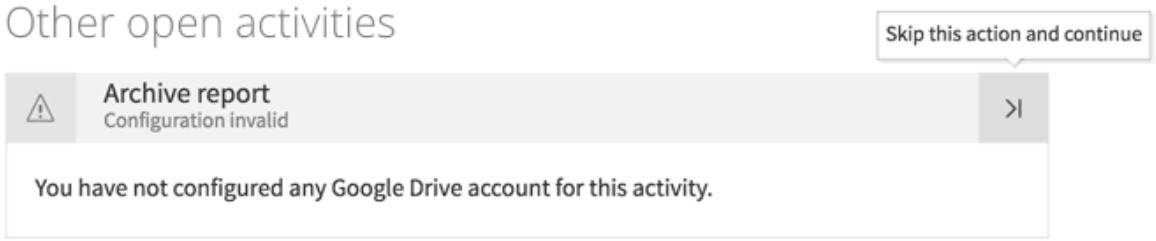


A pending intermediate timer event in the other activities list

To skip the intermediate timer event, use the *Skip timer* button to the right of the timer name.

3.15 Skipping failed tasks

During case execution, an automatic task might fail to execute because it has an invalid configuration. A *Google Drive - Upload file* (page 64) task will fail if you don't configure a Google account, for example. You can manually skip some kinds of failed task, so that case execution continues.



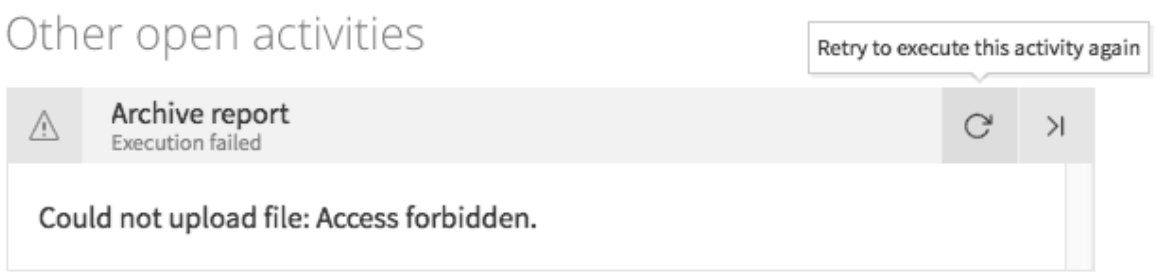
An option to skip a task that failed to execute

To skip a failed task, use the *Skip task* button to the right of the task name in the case's task list.

On the *Skip action* dialogue that opens, enter a reason to explain the decision for skipping the task. The history panel will show the reason with the task completion event, so other people will know why you skipped the task.

3.16 Retrying failed tasks

An automatic task might also fail for an external reason that you can resolve. A *Google Drive - Upload file* (page 64) task will fail if the configured Google account doesn't have permission to write to the selected Google Drive folder, for example. You can manually retry some failed tasks, after resolving the external issue, so that task completes successfully and case execution continues.



An option to retry a task that could succeed on the next try

To retry a failed task, use the *Retry task* button to the right of the task name in the case's task list.

3.17 Reusing start information

Note: This feature cannot be used with public forms.

When you're filling out a form (such as when requesting time off from work), sometimes you may want to start several cases re-using the same data, to avoid having to manually fill the form with the same information each time. There are two ways to do so:

- Fill out and start a case with a form. If the form you filled out has a confirmation message, you will see the *Reuse start info* button. Click it to be brought to the start form page with the same pre-filled form fields.
- In the case view, click the *Reuse start info* button in the case details view. You will be brought to the start form page, where your form fields will be pre-filled with your previous data.

Chapter 4

Processes

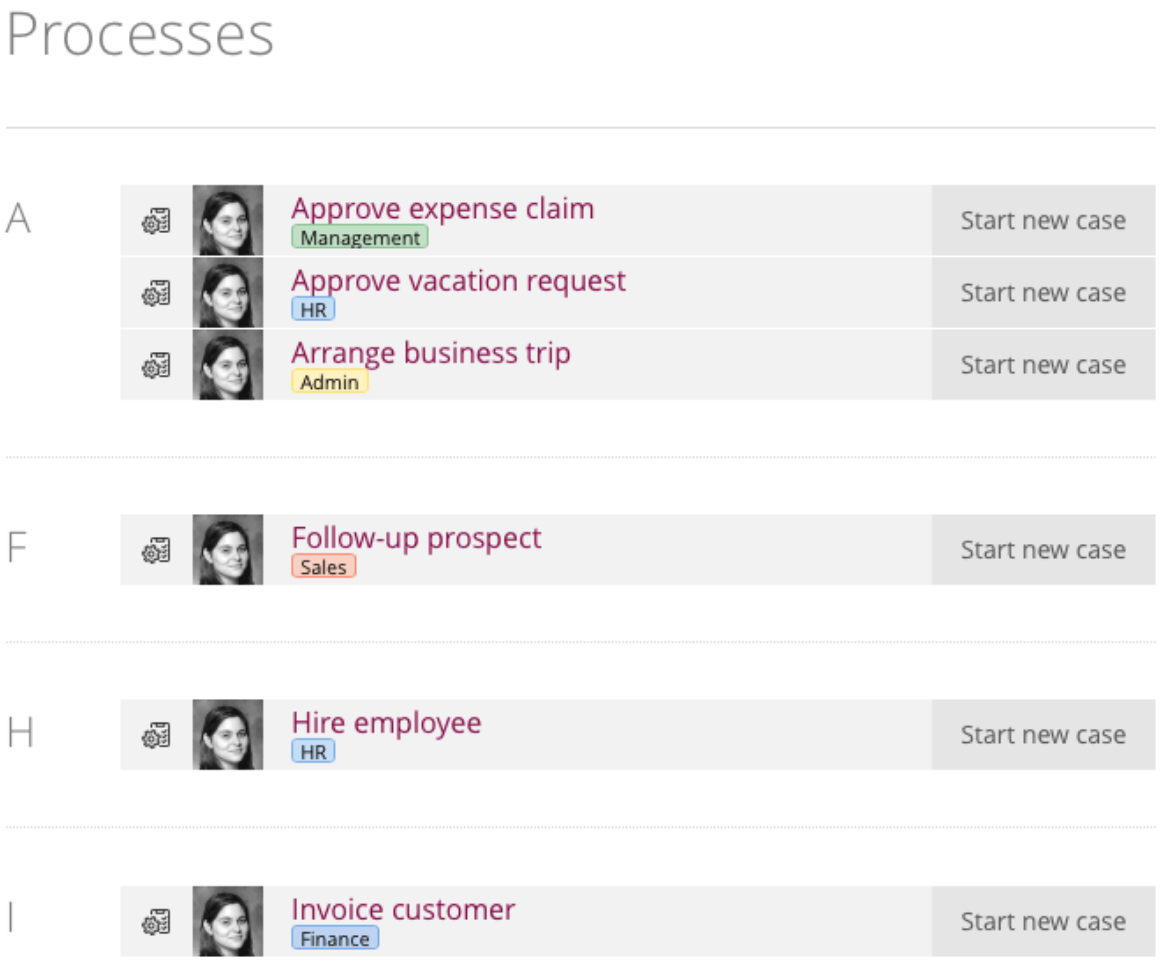
A process defines a template for automating repetitive work, like a recipe that describes the actions that you perform to achieve a goal. For example, consider a Hire employee[9] process. Each time an organization hires someone, the recruitment team has to complete a number of tasks, including 'Evaluate CV', 'Plan interview' and 'Interview candidate'. Each time someone starts the process, Workflow Accelerator creates a new case.

Use the process builder to create and configure executable processes. You can think of an executable process as a kind of software, but you will find it easier to build automation using processes. With Workflow Accelerator, non-technical people can create useful processes.

4.1 Browsing processes

Select *Processes* in the main menu to browse your organization's processes. Each process has an icon that indicates what kind of *trigger* (page 30) it has, the process owner's avatar and the process name. You can also add labels to categorize processes. If you have published a process, you can use the button to *start a new case* (page 14).

[9] https://www.signavio.com/workflow-examples/hire-employee/



4.1.1 Filtering the processes list

To make it easier to browse a long processes list, you can filter the list so it only includes the processes you want. You can use a combination of several filters, to limit the list to processes that match the selection.

- *Labels* - select one or more labels
- *Owner* - select a user
- *Trigger* - select a trigger type: *Email*, *Form*, *Manual*, *Salesforce* or *Signavio Approval Workflow*
- *More Filters* - select a publication status: *Published* or *Unpublished*

To remove a filter, select it again.

4.2 Creating a process

To create a process, select *Processes* in the main menu, then the *Create new process* button. In the text input field, enter a process name.

4.2.1 Choosing a good process name

Use the following guidelines to choose a good process name, to make the list of processes easier to read and talk about.

1. Describe the process goal.
2. Use an imperative verb phrase that completes a sentence like *For your next job, you have to...*
3. Use more than one word, to get a descriptive name.
4. Avoid using more than three or four words.
5. Avoid vague words like "manage", "do", "process" or "handle".

Hire employee, for example, summarizes a process better than *Recruitment*.

If you group or annotate process names by adding prefixes or suffixes, consider using *Labels* (page 30) instead.

4.2.2 Next steps for a new process

Once you have created a process, continue to build the process model using the following process builder sections.

- *Triggers* (page 30) determine how you start new cases for the process
- *Actions* (page 35) define the flow for tasks and other steps in the process
- *Details* (page 39) include process metadata and access control.

4.3 Labels

When several departments in your organization create processes, the *processes list* (page 28) becomes full of other people's processes. Labels categorize processes so you can filter the list by label.

After creating a process, select *Click to add labels* to choose one or more labels from a list. The list starts with a set of default labels that administrators can *configure* (page 118). Select a label's delete icon to remove it from a process.

4.4 Triggers

A trigger in a process specifies how the process starts. Triggers do not have any relation to *start events* (page 104).

4.4.1 Manual trigger

A manual trigger gives you the simplest way to start a process. With a manual trigger, you start processes manually in Signavio Workflow Accelerator, by selecting “Start new case” and then selecting the process to start.

4.4.2 Form trigger

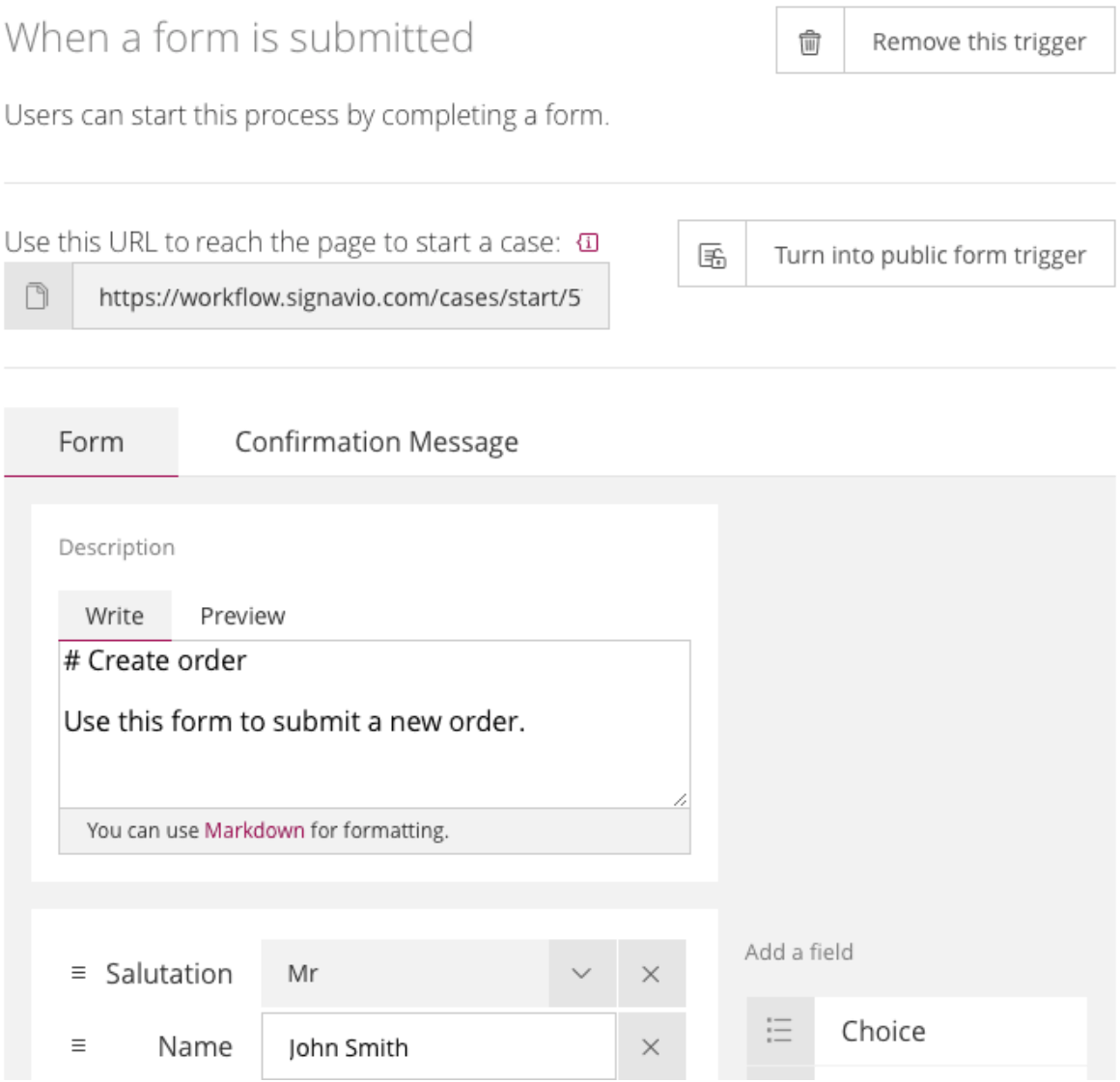
Form triggers allow you to start a process by using a form.

There are two kinds of form triggers: public and private.

- **Public form triggers** allow anyone, including people outside your organization without user accounts, to start processes in your organization via a form. Public forms do not have user or connector fields, for data security reasons. When a private form containing those fields is made public, Signavio Workflow Accelerator will automatically remove them from the form. Cases that are created via a public form trigger will not have a case creator listed, even if they were created by a logged in user.
- **Private form triggers** allow only registered members of your organization to start process via a form.

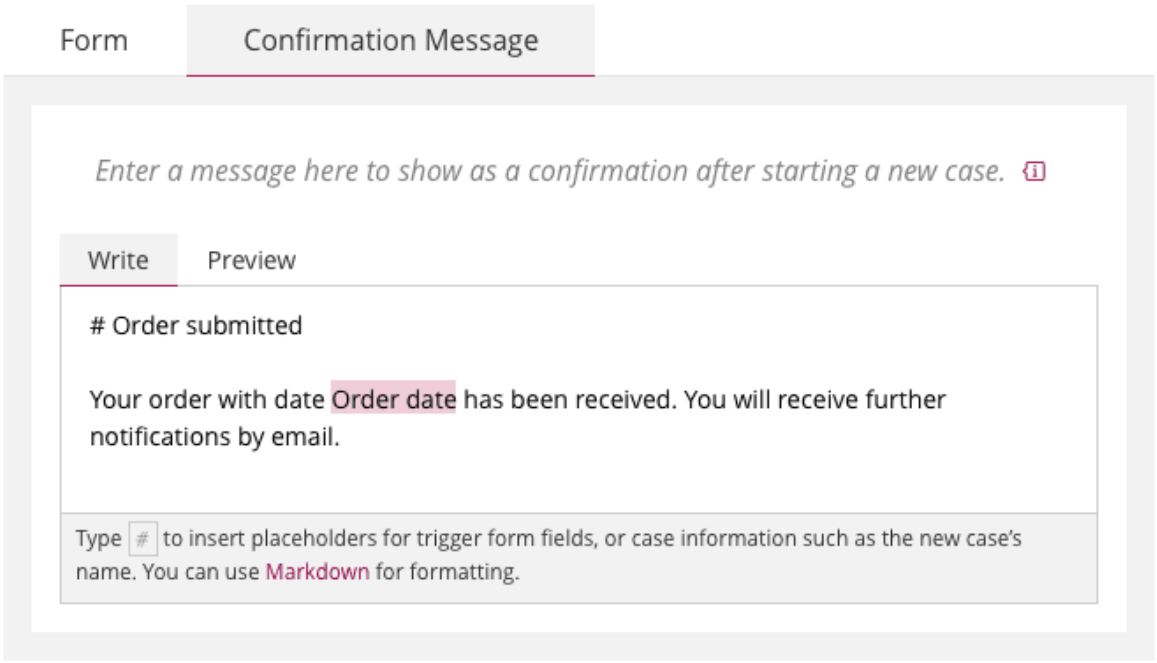
You can activate and deactivate the public form trigger by clicking the *Turn into public form trigger* button and publishing a new version of the process. (The *Versions* tab will record if a public form was used in any of the previous versions of a process.)

Regardless of the type, after selecting a form trigger, use the *form builder* (page 85) to specify form fields.



Form trigger configuration - using the form builder to define a trigger form

For some processes, such as an HR request from an employee, the person who starts the case doesn't have access to view the case. This means that after using a form trigger to start a case, they don't see the case details view, and might not know that the case started successfully. For these processes, you can use the *Confirmation Message* template to show a message to the case creator.

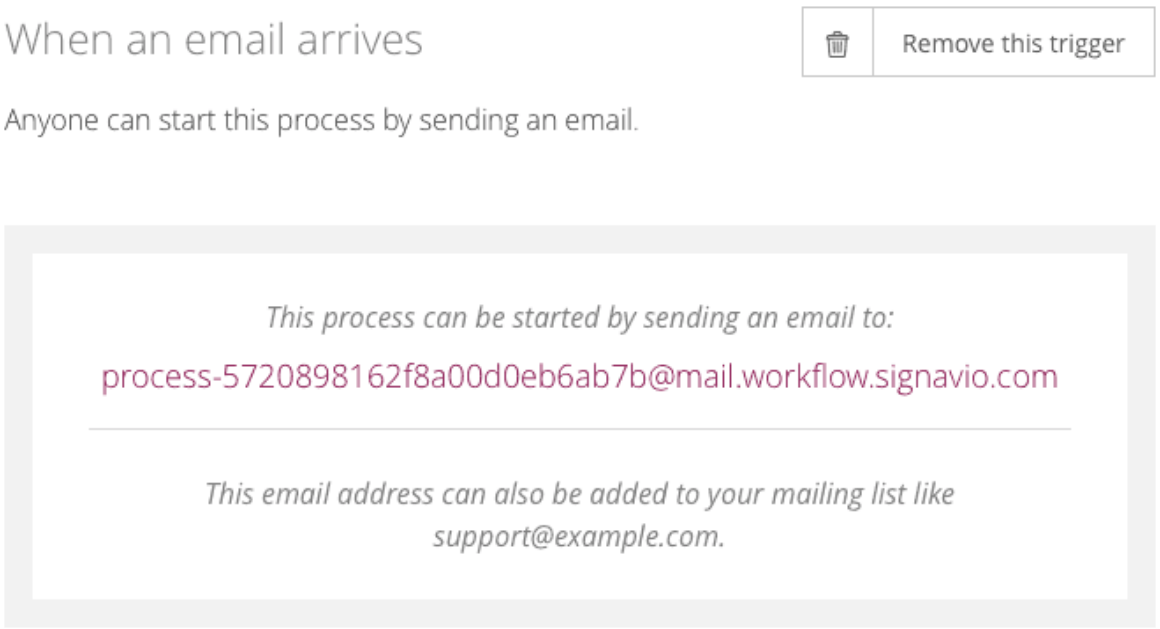


Form trigger confirmation - shown to the case creator after starting a case

In the template, you can use placeholders to insert trigger form field values. If you do not define a confirmation message, you will see a default message informing you that the case is closed and you don't have permission to see it.

4.4.3 Email trigger

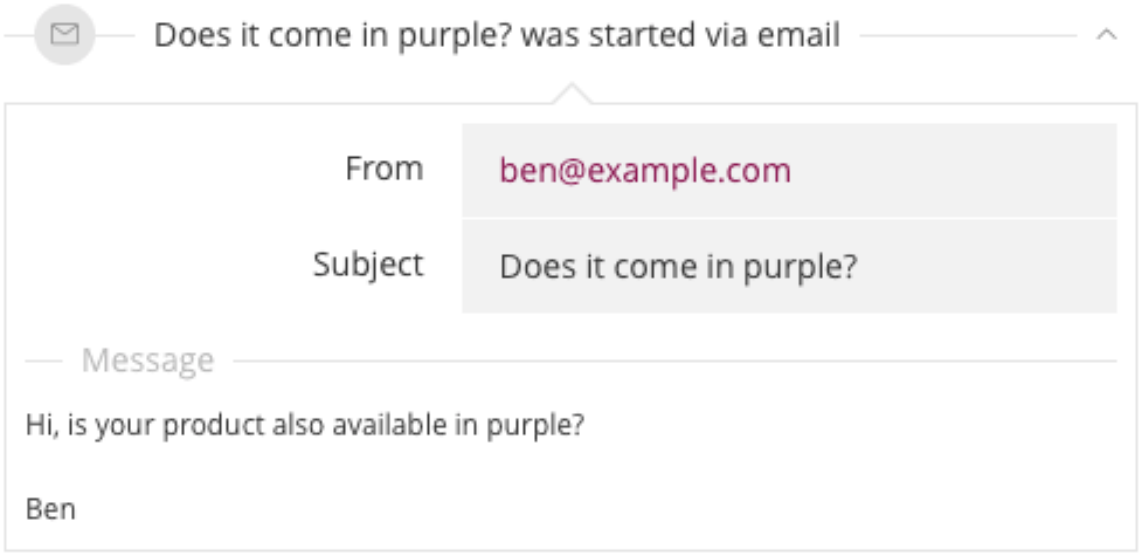
An email trigger starts a new case for each email that you send to the trigger's Workflow Accelerator email address. Note this differs from reading an existing email account, such as your own. After selecting the email trigger, you can see its email address:



Mail trigger configuration

The email trigger creates a *Trigger email variable* (page 124).

You can use an email trigger by adding the trigger email address to a mailing list, such as *support@example.com* or *info@example.com*. You'll have to ask the administrator of the mailing list to add the process trigger's email address to the list. Once you have done this, the process trigger address will also receive any email sent to the mailing list, starting the process in Signavio Workflow Accelerator. You will then see the email in the history panel:

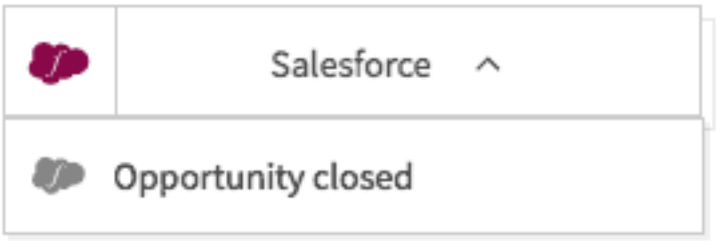


Trigger mail in stream

4.4.4 Salesforce trigger

A Salesforce trigger starts a new case in response to Salesforce sending an outbound message as part of a Salesforce workflow. Before you can use a Salesforce trigger, configure *Salesforce Integration* (page 161).

When you have configured a Salesforce service, you can select it as a process trigger, so that messages from Salesforce will now trigger new cases.



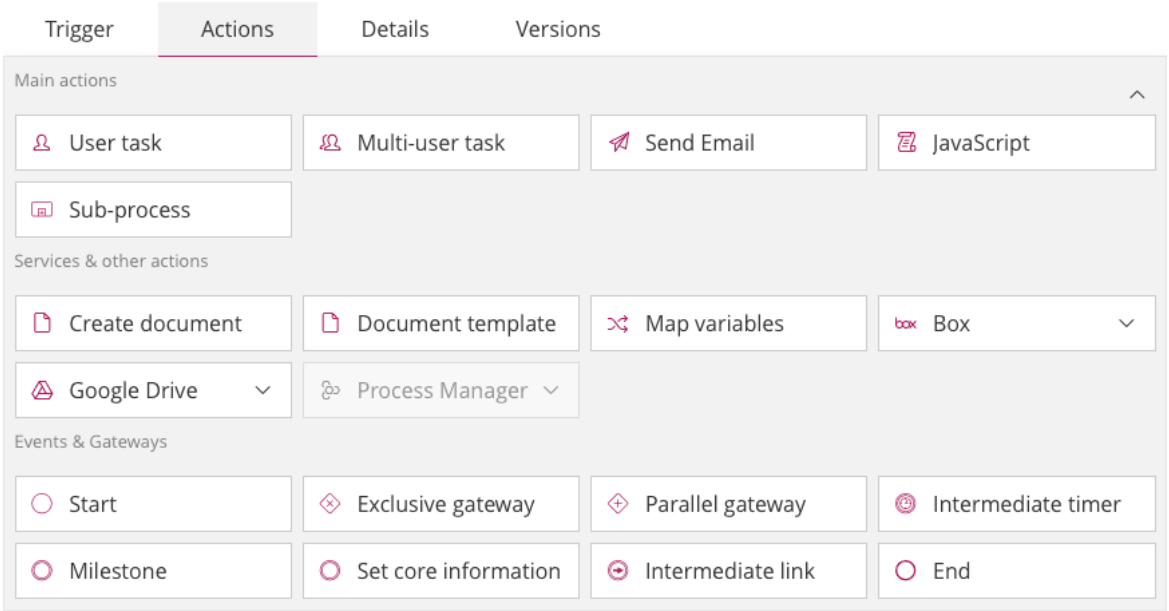
Example of a Salesforce trigger

In the process definition, you can use the Salesforce object fields from the Salesforce message just like normal *variables* (page 43).

4.5 Actions

Actions represent the steps in a process - things to do. Actions include things like user tasks in Signavio Workflow Accelerator, operations on a file in a document management system or any other actions that represent work that someone will perform as part of a process. When starting a process, Signavio Workflow Accelerator will execute the actions in a process in the proper order. The process *control flow* (page 97) determines this ordering, using transitions, gateways and events.

A process can include different *Action Types* (page 56). A user task will create a task in a case. A "Send Email" action will send an email. A "Google file upload" action will upload a file to a Google Drive folder.



The Process builder's *actions palette*

The BPMN[10] diagram editor shows actions and control flow elements, such as events and gateways. Use the diagram editor to add sequential flows between actions, decisions and other control flow behavior.

4.5.1 Adding a transition

A transition specifies sequential flow, which means the next action only starts when someone has completed the previous one.

To add a transition, click to select the first action. Several symbols appear to the right of the selected element:

Start creating a transition by dragging the transition symbol to the destination

Click the transition symbol  and drag it to the destination element. When you drag the symbol over the destination element, it indicates that you can drop to create the transition:

Creating a transition hovering over destination

Release the mouse button over the destination to create the transition.

[10] http://en.wikipedia.org/wiki/Business_Process_Model_and_Notation

4.5.2 Creating the next user task

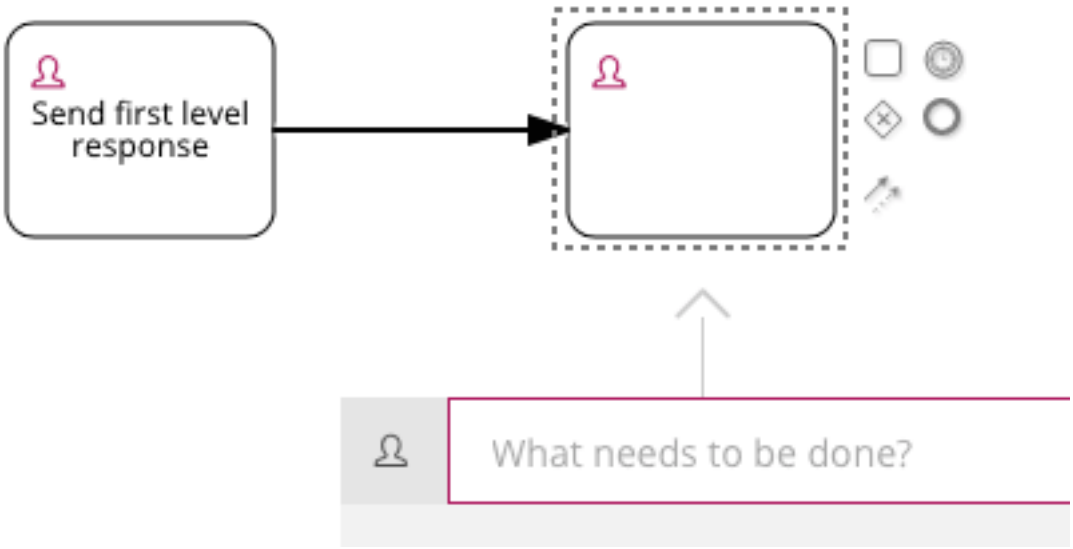
You can easily create the next User Task in a process in the same way you created transitions, above. Start by selecting the previous action:

Start creating the next user task by dragging the action symbol

Click the rounded rectangle symbol  and drag it to an empty place on the canvas.

Drag the rounded rectangle symbol to an empty place

Drop the symbol where you want to create the next user task. Release the mouse button to create new user task where you dropped it, with a transition from the previous action.



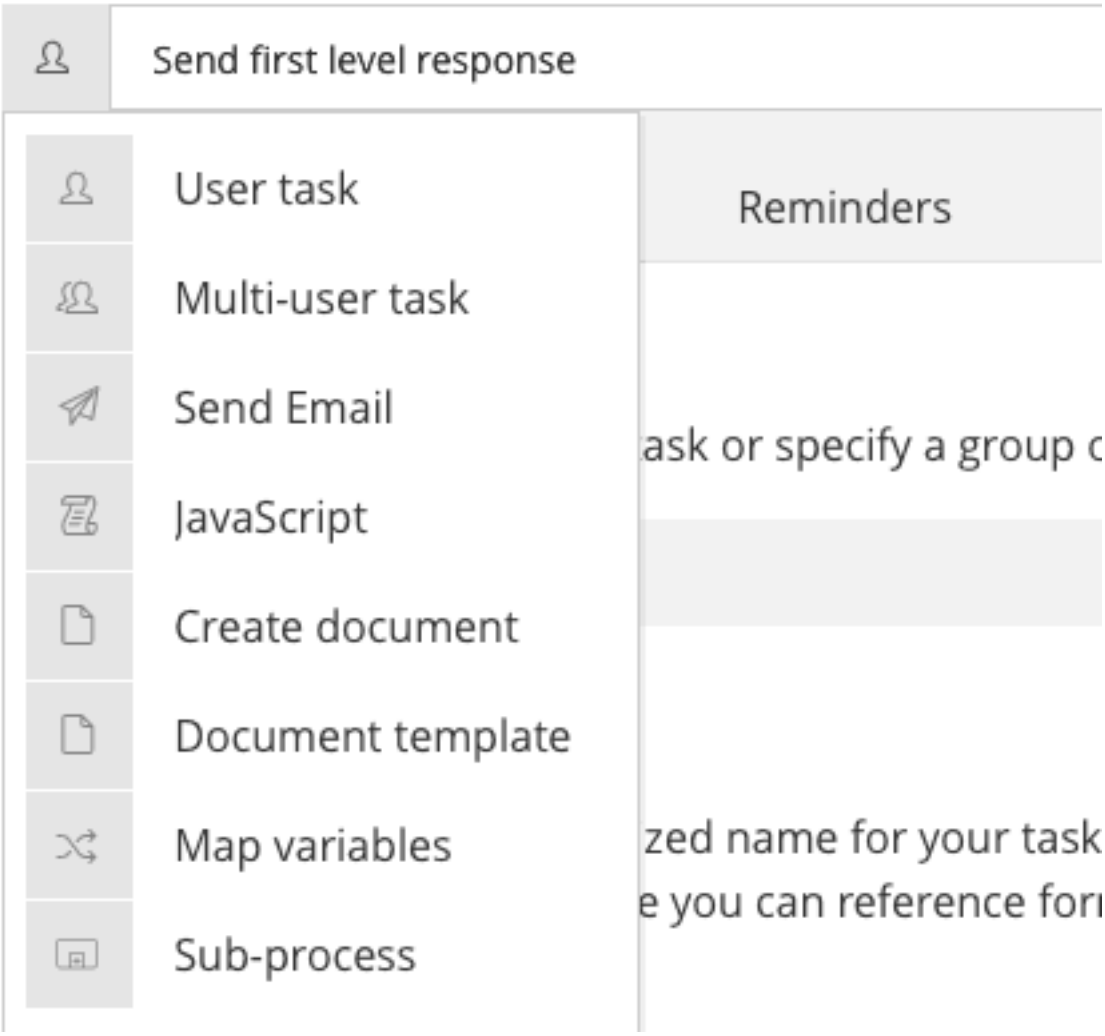
Adding the next user task and its transition at the same time

Alternatively, just click the user task icon to create a new user task in the default location, with a transition.

4.5.3 Changing the action type

Sometimes, modifying a process means changing an action from one type to another. In the recruitment process, you might change the action to reject the candidate from a manual task to an automatic email task.

To make this change, first select the action in the process editor to open the configuration panel, then click the action type icon at the top-left corner of the configuration panel to open the list of action types. Next, select *Send Email* from the list to change the action type.



Changing a User task (page 56) to a Send email (page 62) task.

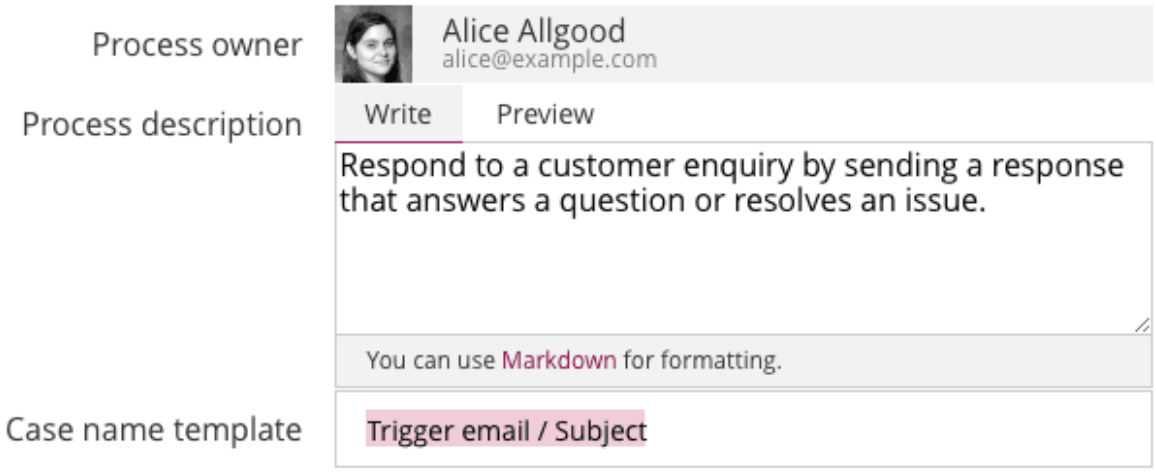
Warning: Changing the action type discards the previous action type's configuration, such as a user task form or an email template. If you change the action type back, the editor will not restore the original configuration.

4.5.4 Control flow elements

The diagram also includes control flow elements, such as events and gateways. Unlike actions, control flow elements don't represent something that should happen. Instead, you use events and gateways to specify the flow between the actions.

4.6 Details

In the process builder, select the *Details* tab to further configure the process.



Configuring process details

On the *General* tab, configure the following process properties.

- *Process owner* - shown on the *Processes* page to indicate who has responsibility for a process model, and used as the default recipient of some *notifications* (page 120).
- *Process description* - documents a process, usually by describing the process goal.
- *Case name template* - the name for new cases of this process, usually containing trigger *variables* (page 43) so that each case has a different name.

Use the *Access control* tab to restrict access to this process and its cases.

Use the *Field overview* tab to view and rename this process' *variables* (page 43).

Use the *Core information* tab to view and edit important information about a process or case.

4.7 Versions

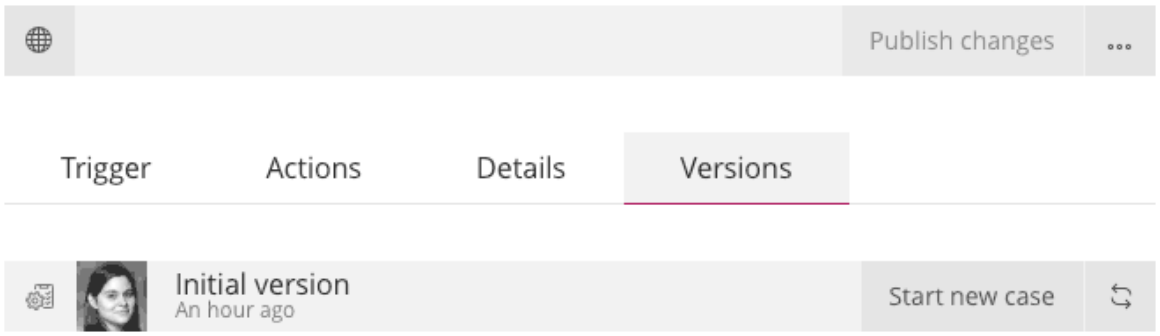
When you use the process editor to edit your process model, Workflow Accelerator saves all of your changes immediately. You can go back and edit the process again later, and it will not have changed. However, to execute a process by starting a new case you need a published version.

4.7.1 Publishing a process version

The process editor's "Versions" tab shows a list of published versions. Until you publish a version of a process, the following message is displayed:

The process editor's "Versions" tab with no published versions

You can only start a new case for a process that has a published version, hence the light green button displays "Publish to run this process". After you publish the first version, the list shows version #1 and you can start a new case using that version.

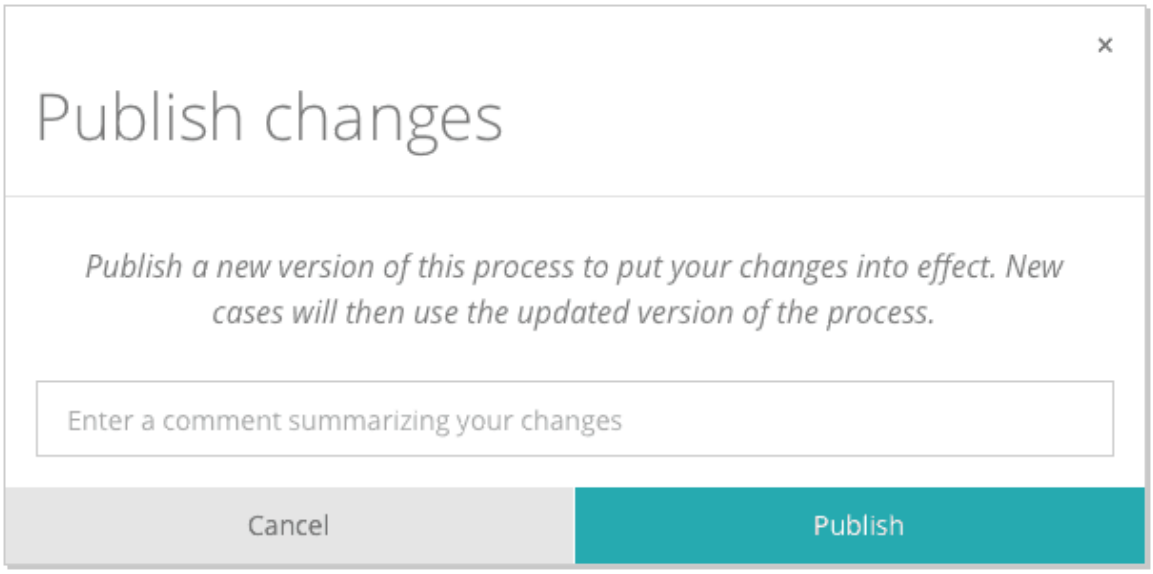


The Versions *list after publishing the first version*

Version #1 always has the description "Initial version". For later versions, you can add your own description.

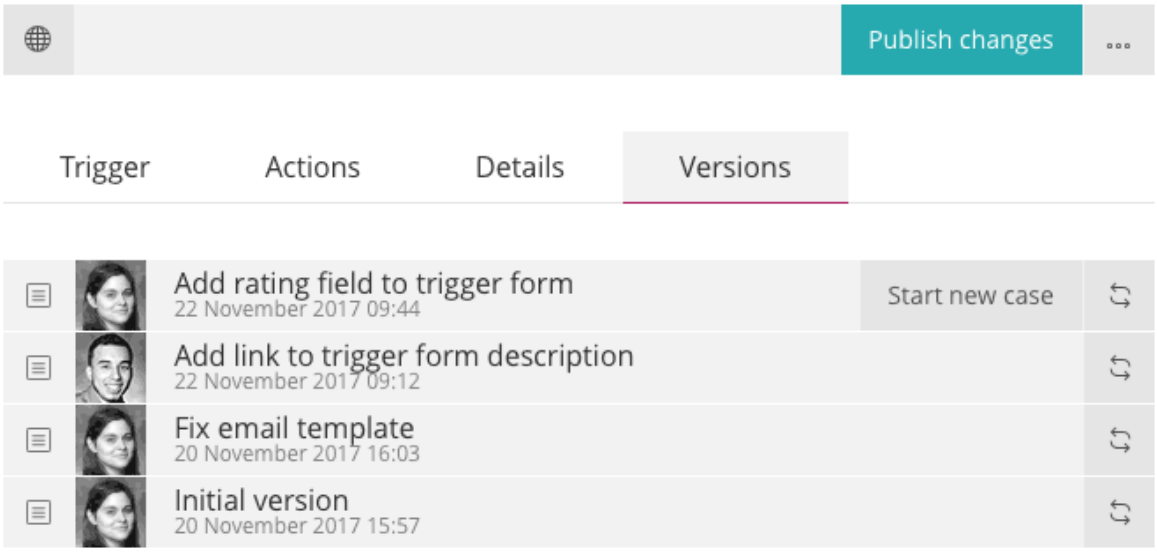
4.7.2 Adding version comments

After the first published version, you can add a comment to describe the changes when you publish a new version of a process.



The Publish changes prompt, where you can add an optional version comment

You can omit the version comment, but it helps collaboration between team members by making process modeling more transparent.



Published versions with descriptive version comments

In this example, each version has a short comment that describes the changes.

4.7.3 Writing good version comments

When you write version comments, use the following tips to make them more consistent and useful.

- Make the comment an imperative phrase that starts with a word like “Add” or “Fix”.
- Capitalize the first word and don’t include a full-stop at the end, for consistency.
- Describe specifics, instead of vaguely referring to “changes”.

- Keep it short; 3-10 words usually suffice.
- Consider making the comment longer to explain why you made this change.

You may find it easier to publish a series of small changes, creating a number of intermediate versions instead of one big change. Fine-grained versions make the version history more useful.

4.8 Variables

Variables contain the workflow data that the process defines. You can use variables in a case name template and when configuring the output of some action types. For example, you can use variables to repeat workflow data on a *user task form* (page 84), or include a variable value in an *email task* (page 62) subject line or body text.

These variables contain all of the information from forms as well as information required by the process actions. Each case stores its own values for each workflow variable.

You will usually add a variable to your workflow by adding a *form field* (page 87). You can also create variables in a *JavaScript action* (page 74), to capture data that the script retrieves or calculates.

A trigger form that populates Name *and* Date of birth *variables for use in a workflow*

In addition to your own workflow variables, Workflow Accelerator automatically creates variables that give you access to additional data in each case. The *Case variable* (page 123) contains data from when Workflow Accelerator creates the case. An *Email trigger* (page 33) adds an *Email variable* (page 124) that contains the trigger email.

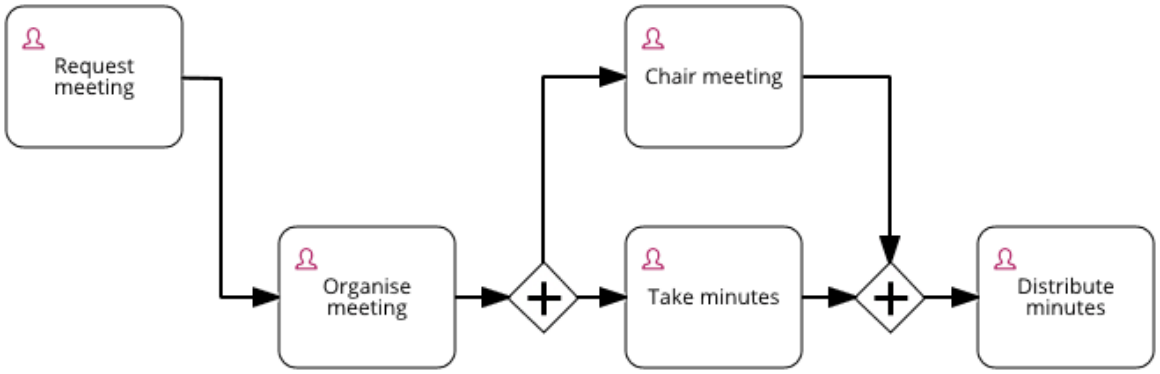
Variables can have different *Data types* (page 124) that determine which kind of data the variable stores, such as text or a date, and whether the data has a single value or contains multiple fields.

4.9 Roles

Creating a role gives a process-specific name to whoever performs one or more process tasks. You can optionally configure a role with a list of candidates. Roles have the same function as swimlanes in BPMN.

Process roles differ from organizational roles. A process role only lasts for the duration of a case, while organizational roles last longer and relate to the job you perform at the organization. For example, when you have a meeting, one person sometimes takes the role of chairperson. That person doesn't have the job title *Meeting chair* - they've just adopted that role for the duration of the meeting.

A process in Signavio Workflow Accelerator can define roles, in the same way that a business meeting 'process' has roles for whoever chairs the meeting (the 'Chair') and whoever takes minutes (the 'Secretary'). The following meeting process model assigns the tasks on the top row to the *Chair* and the tasks on the bottom row to the *Secretary*.



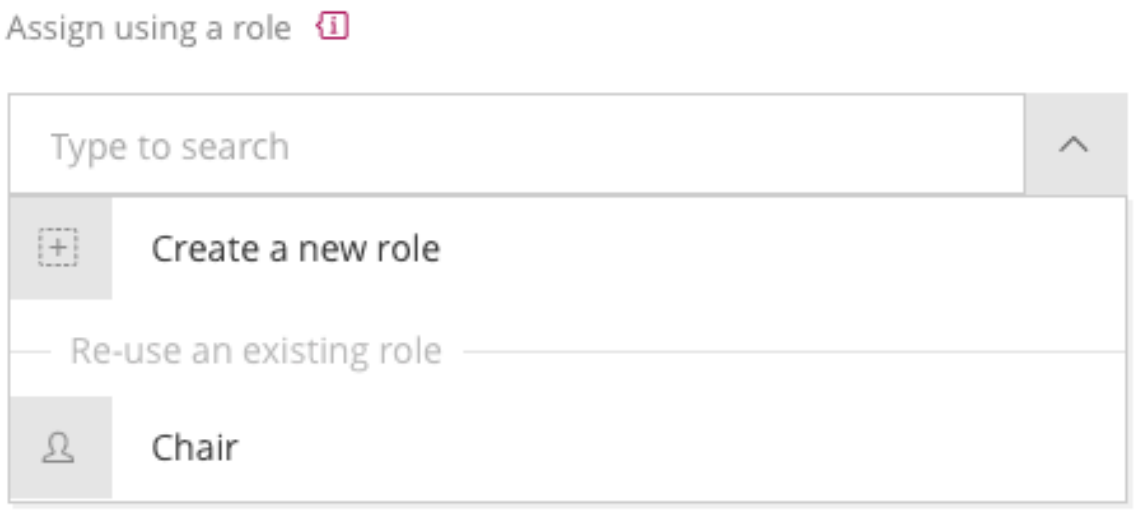
A Meeting *process, with tasks for* Chair *(top row) and* Secretary *(bottom row) roles*

In each meeting (each *case* (page 13) in Signavio Workflow Accelerator), one person takes the role of chair, and one the role of secretary. These assignments generally don't change during a meeting. Similarly, Signavio Workflow Accelerator role assignments don't change during a case. Signavio Workflow Accelerator automatically assigns each new task with a role assignment to the person who already has the role.

In Signavio Workflow Accelerator, a process role works like a workflow variable that you use to assigning tasks. A role variable has the *User* (page 127) type and stores a single user.

These process roles differ from organization roles. For example, you can have the *Team lead* role in your organization, an assignment that does not necessarily have an end date. A process role, such as *Meeting chair*, has a different scope and only applies for the duration of a single case.

To assign a role to a user task, open the task's configuration panel, select the *Assignment* tab, and use the *Assign using a role* menu on the right-hand side.

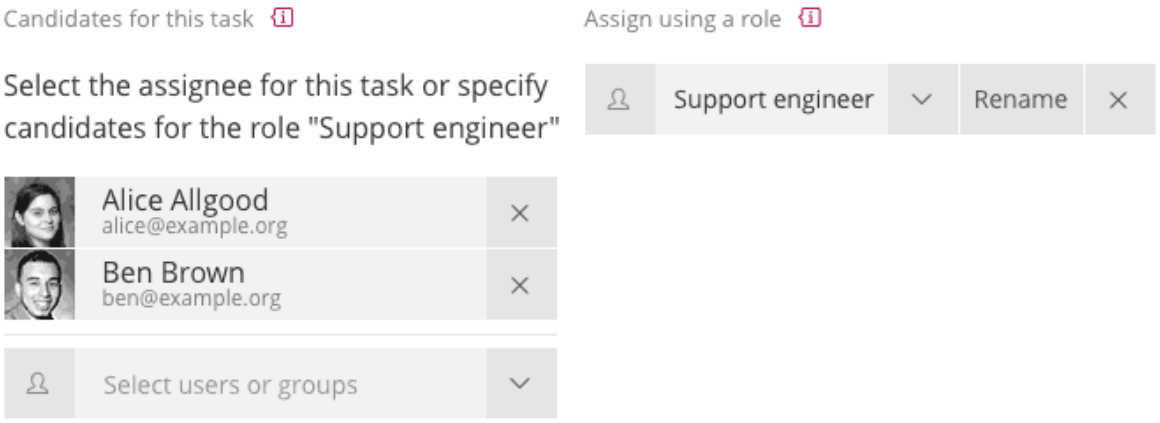


To assign a role, use the menu to create a new role or select an existing role

You can also use the edit icon next to the role name to rename the role.

4.9.1 Role candidates

You can use a role to assign multiple tasks a person from a group of candidates. For example, you might have a support process that includes three user tasks that you assign to a support engineer.



Assigning a task to a Support engineer *role with candidates Alice and Ben*

Instead, assign the three tasks to a new role called “Support engineer”, and add the relevant people as candidates for the role.

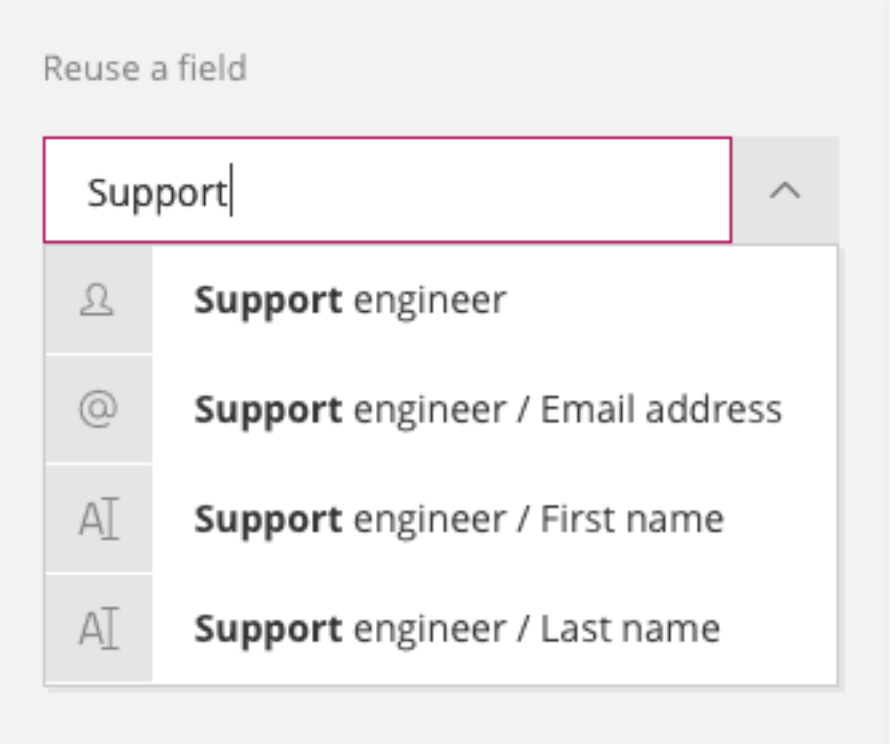
Creating the first task that has a role will notify all of the candidates for the role. When one of the candidates takes the task, Signavio Workflow Accelerator will assign the subsequent tasks with the same role to the same person. That helps this person work more efficiently because they have the context knowledge about that case.

If you reassign a task that has a role assignment, Signavio Workflow Accelerator will update the role variable, and assign all subsequent tasks with the same role to the new assignee.

4.9.2 Using a form field to assign a role

When you execute a process, you normally assign a specific person to a role by using the assignee button to select someone. Sometimes, you want this assignment to an explicit part of the process, to make sure the assignment happens at the right time. For example, you may find it important to assign the *Support engineer* before completing an *Initial investigation* task.

You can do this by adding the role assignment to a form, because you can use task assignment roles as process variables, just like any other *User* form field.



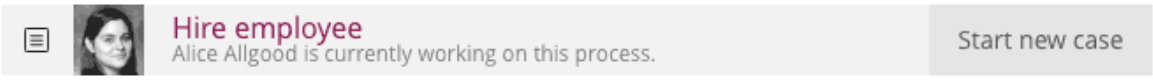
Adding the Support engineer *role assignment to a form*

To add a role assignment to a form, first define the process role, such as the *Support engineer* role created above, then on the form, under the *Reuse field* heading, select the role variable to add it to the form.

Alternatively, you can first define the *Support engineer* role by adding a field with type *User* to the *Initial investigation* task's form, and then select the *Support engineer* role on another user task's *General* configuration.

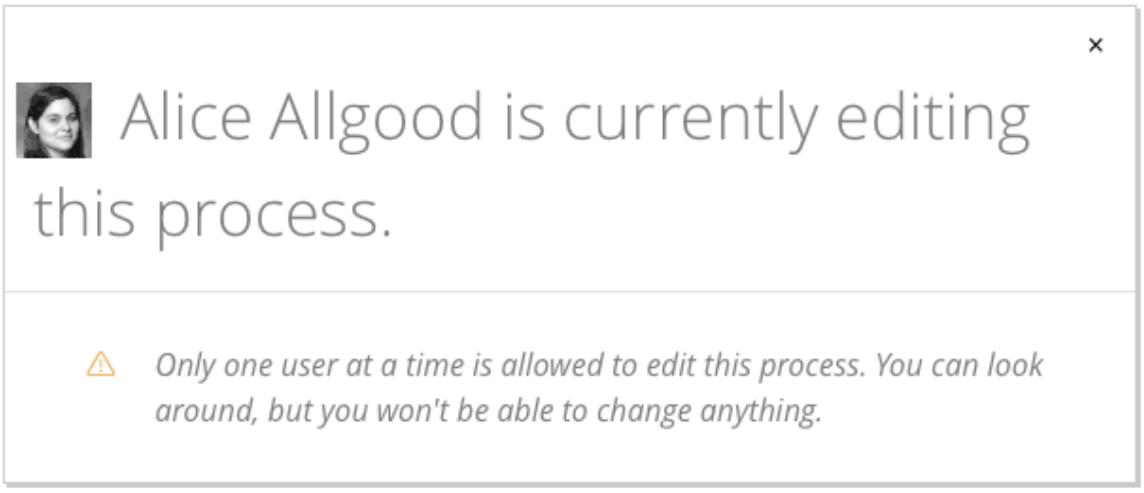
4.10 Process locking

In the process builder, only one person can edit a process at a time. While someone continues to edit a process, the "Processes" list shows a message.



The "Processes" list message while someone else edits the process

You can still open the process, but you will see a warning message explaining that you cannot make changes:



Warning that you cannot edit a process at the same time as someone else

You can edit the process and make changes after the other person leaves the process editor, by opening the “Tasks” list, for example.

4.11 BPMN import

You can import a Workflow Accelerator process model from a BPMN 2.0 XML file. You can use this to import a model that you created in another tool, or to load a file that you saved using the *BPMN export* (page 48) option.

To import a process model, on the “Processes” page, click the “Import BPMN” button and select the BPMN XML file.

The “Import BPMN” button on the “Processes” page

Workflow Accelerator does not support all BPMN 2.0 elements, so the process may appear differently in Workflow Accelerator. The following table lists supported BPMN elements, and the corresponding action type.

Supported BPMN elements

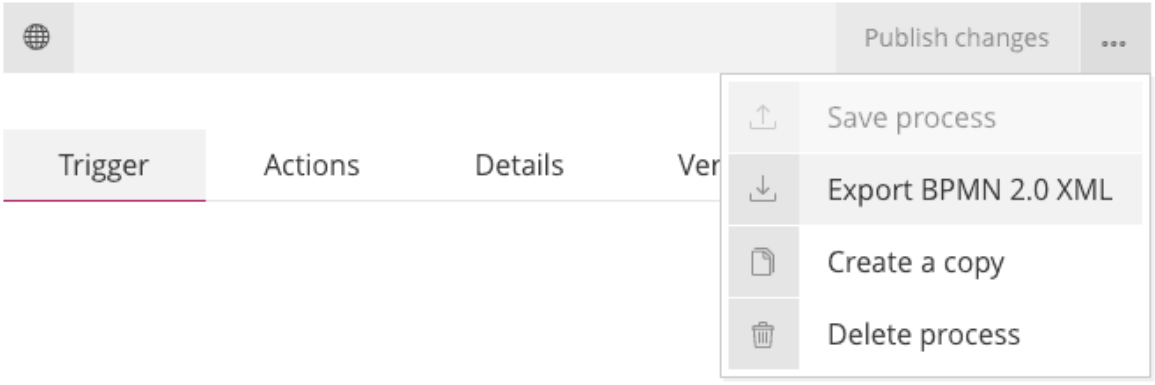
BPMN element	Action type
Business rule task	*DMN Rule Task* (page 77)
Call activity	*Sub-process* (page 75)
End event	*End event* (page 104)
Exclusive gateway	*Exclusive gateway* (page 97)
Intermediate timer event	*Intermediate timer event* (page 105)
Manual task	*User task* (page 56)
Parallel gateway	*Parallel Gateway* (page 101)
Script task	*JavaScript Integration* (page 134)
Send task (`type=email`)	*Send email* (page 62)
Service task (`type=changeState`)	Signavio Process Manager state change
Service task (`type=boxFileUpload`)	*Box Upload file* (page 72)
Service task (`type=googleAddCalendarEvent`)	*Google Drive - Add calendar event* (page 70)
Service task (`type=googleCloudPrint`)	*Google Drive - Print file* (page 67)
Service task (`type=googleDriveAddRow`)	*Google Drive - Add row to sheet* (page 69)
Service task (`type=googleDriveFileUpload`)	*Google Drive - Upload file* (page 64)
Start event	*Start event* (page 104)
Sub-process (embedded)	*Sub-process* (page 75)
Swim lanes	*Roles* (page 43)
Task	*User task* (page 56)
User task	*User task* (page 56)

Workflow Accelerator removes unsupported elements, such as message events.

4.12 BPMN export

You can export a Workflow Accelerator process model as a BPMN 2.0 XML file. You may find this useful for opening the model in another tool that supports BPMN, or to make a backup that you can load using the *BPMN import* (page 47) option.

To export a process model, first open it in the Workflow Accelerator process editor. Select the menu next to the "Publish changes" button, then select *Export BPMN 2.0 XML*.



Using the "Export BPMN 2.0 XML" option to save a process model in a file

This results in a file download that describes your process model in BPMN format. Workflow Accelerator only exports one kind of BPMN file, so it doesn't give you any options to configure.

4.13 Copying & deleting processes

You can create a copy of a process or delete it using the process menu shown in the previous section.

The “Create a copy” option duplicates the process in the same organization. You may find it useful to duplicate a process if you want to experiment with changes without publishing changes to a live process. You may also want to duplicate a process to model a special case of the process, instead of adding a conditional flow to the standard process.

The “Delete process” option permanently deletes a process. You can also chose to delete all of its cases. When you click “Delete process”, a confirmation dialogue will open. There, you can determine whether you also want to delete all of the process’s cases before continuing.

In Signavio Workflow Accelerator, you cannot currently delete individual cases, which you might want to do if you have created test cases while developing the process. However, you can use “Create a copy” and “Delete process” to duplicate a process and then delete the original. This deletes all of the cases with the original process, and leaves a copy with no cases.

Chapter 5

Analytics (reporting)

When you work in or manage a team, you sometimes have questions about the work in progress or completed work. For example, management decisions about team resources might depend on what work the team completed last month and what work remains incomplete. While day-to-day case work focuses on completing one task at a time, managing a team of case workers demands an aggregated overview of the team's work.

The *Analytics* menu makes it possible to create and share reports that provide these overviews. Each report runs on demand and aggregates a process' cases in tabular and graphical form in the web user interface.

5.1 Creating a new report

To create a new report, select *Analytics* in the menu, and then *Create new report*. Enter a name for your report that describes the data, such as *First quarter sales*. The finish setting up a basic report, use the list labelled "Select a process" to choose one of your organization's published reports.

When you create a report and select a process, the report shows a table of all of the process' cases. This report now appears in the list of reports that you see when you select *Analytics*. By default, only you can see the report in the list.

5.2 Viewing and exporting results

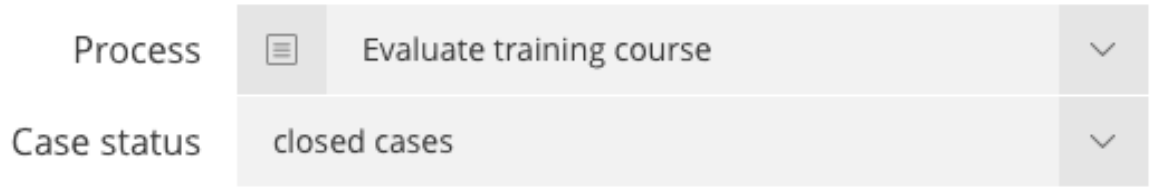
The report page's table shows cases, with one case per row. The table header indicates the number of cases included in the report. Select *Configure columns* to choose which columns the table includes. This works the same way as *Configuring case view table columns* (page 16).

If you have special presentation requirements, you may want to use the same data to produce your own report using different software. To do this, you can download the same data as a CSV file[11] that you can open in spreadsheet software.

5.3 Selecting cases to include

For most reports, you won't want to include all of a process' cases. After all, the *Cases view* (page 14) already shows an overview of all cases. Instead, you can restrict which cases the report includes, by status or by the values of the process' *Variables* (page 43).

[11] https://en.wikipedia.org/wiki/Comma-separated_values



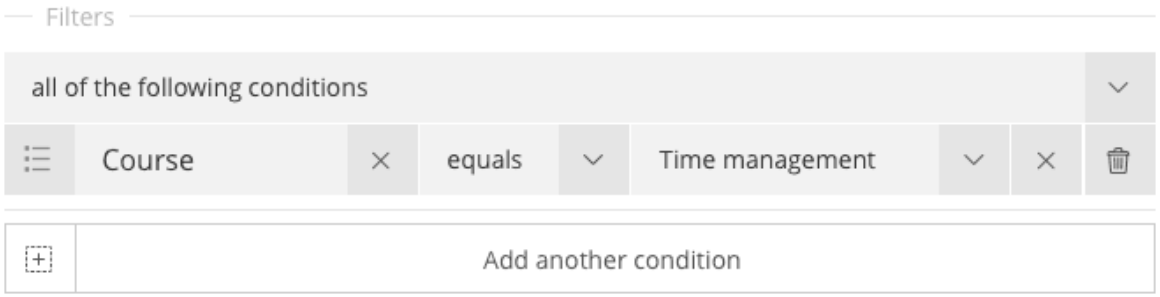
Selecting a process and whether to include open or closed cases

In this example, the *Evaluate training course* process collects ratings for training courses. To create a report that uses these evaluations, select closed cases, which correspond to completed evaluations.

5.4 Excluding cases with filters

Filtering by status lets you create separate reports for completed work, such as a monthly productivity report, and outstanding work. To filter cases by status, use the first pick-list to select between *all cases*, *open cases* and *closed cases*.

To filter cases by variables' values, first select *Add a filter condition*. This adds a field condition, which works the same way as an *Automatic decision* (page 100) condition. Select a field, such as *Case / End date*, a condition, such as *is after*, and a value, such as the last day of the previous month.



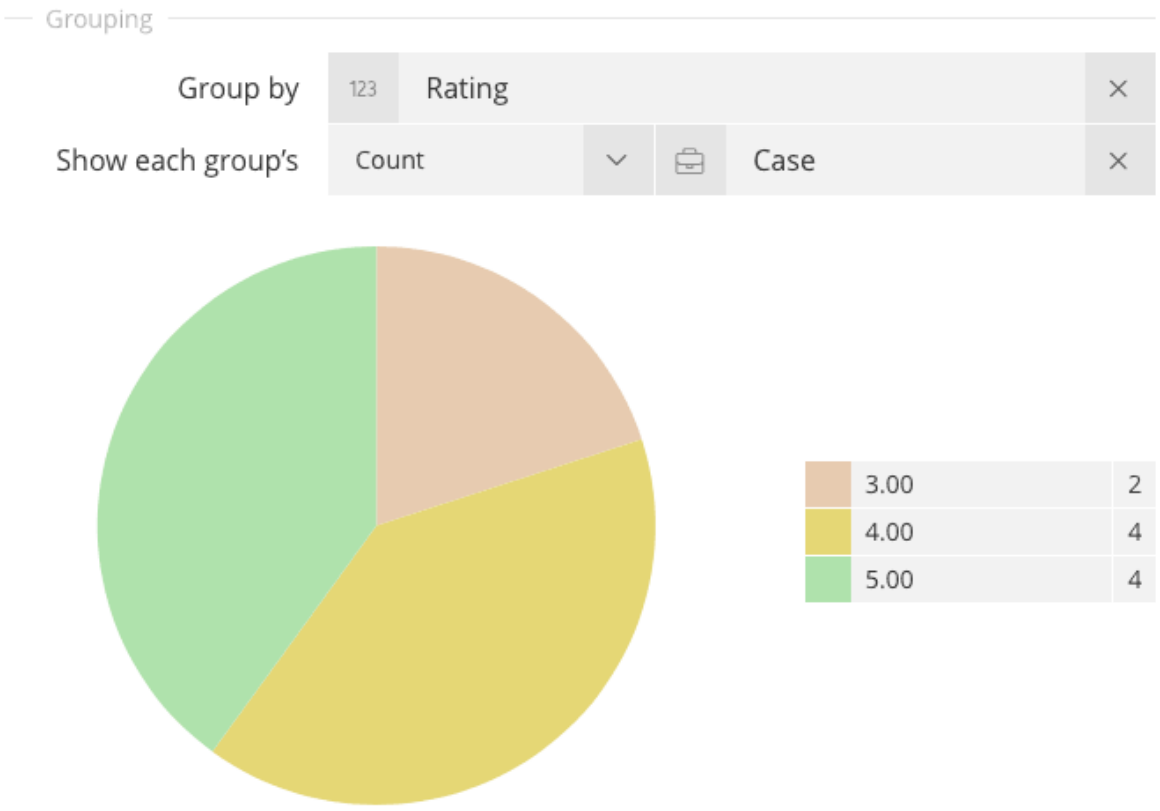
Using a filter to exclude evaluations for courses other than Time management

You can add multiple conditions to further restrict which cases the report includes. A monthly case report would have two conditions for the *Case / End date* field, using the *is after* and *is before* conditions to define a date range.

5.5 Grouping and charts

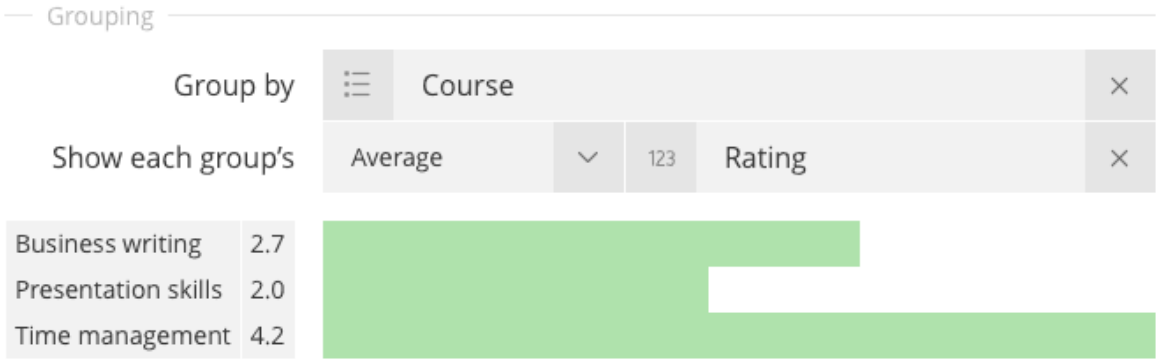
As well as listing cases in a table, you can also group cases by one of the process variables, so that cases with the same value appear together in the table. Grouping cases can also add a visualisation to the report.

To show a pie chart of the different values for a variable, such as the course evaluation ratings, select the *Rating* variable from the *Group by* list. Then select the *Count* of *Case* values to count the evaluations with each distinct rating value.



Grouping evaluations by rating, and counting the number of evaluations with rating 3, 4 or 5

Reports can also calculate *average* values for a numeric field, as well as counting cases. To show a bar chart of the average rating for each course, group by the *Course* (title) and show the *Average* of *Rating* values.

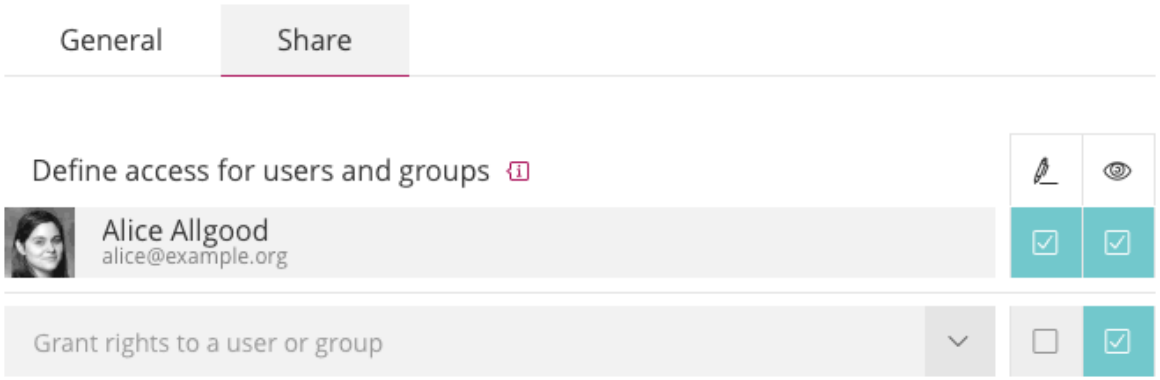


Grouping evaluations by course title, and calculating each course's averge rating

As well as *Average*, you can also show the *Minimum* or *Maximum* for each rating.

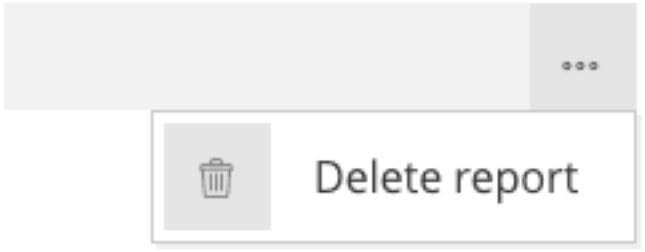
5.6 Sharing a report

To share a report, select the *Share* tab, underneath the report title and next to the *General* tab. From there you can specify users, groups, or organizations that you wish to share your report with.



5.7 Deleting a report

To delete a report, select the *Delete report* option from the top-right menu, under the report title.



Chapter 6

Search

The search function in Workflow Accelerator helps you find specific tasks, cases, processes and reports. Search terms are searched in names and description of the respective category. With the exception of the search for cases: here, only the name is taken into account.

How to search in Workflow Accelerator:

- Wildcards are not supported.
- With search terms in quotation marks, you can search for a coherent string.
- Exclude search terms by a - (minus) before the search term.
- Upper/lower case is not taken into account.
- You must enter at least three characters for the search term.

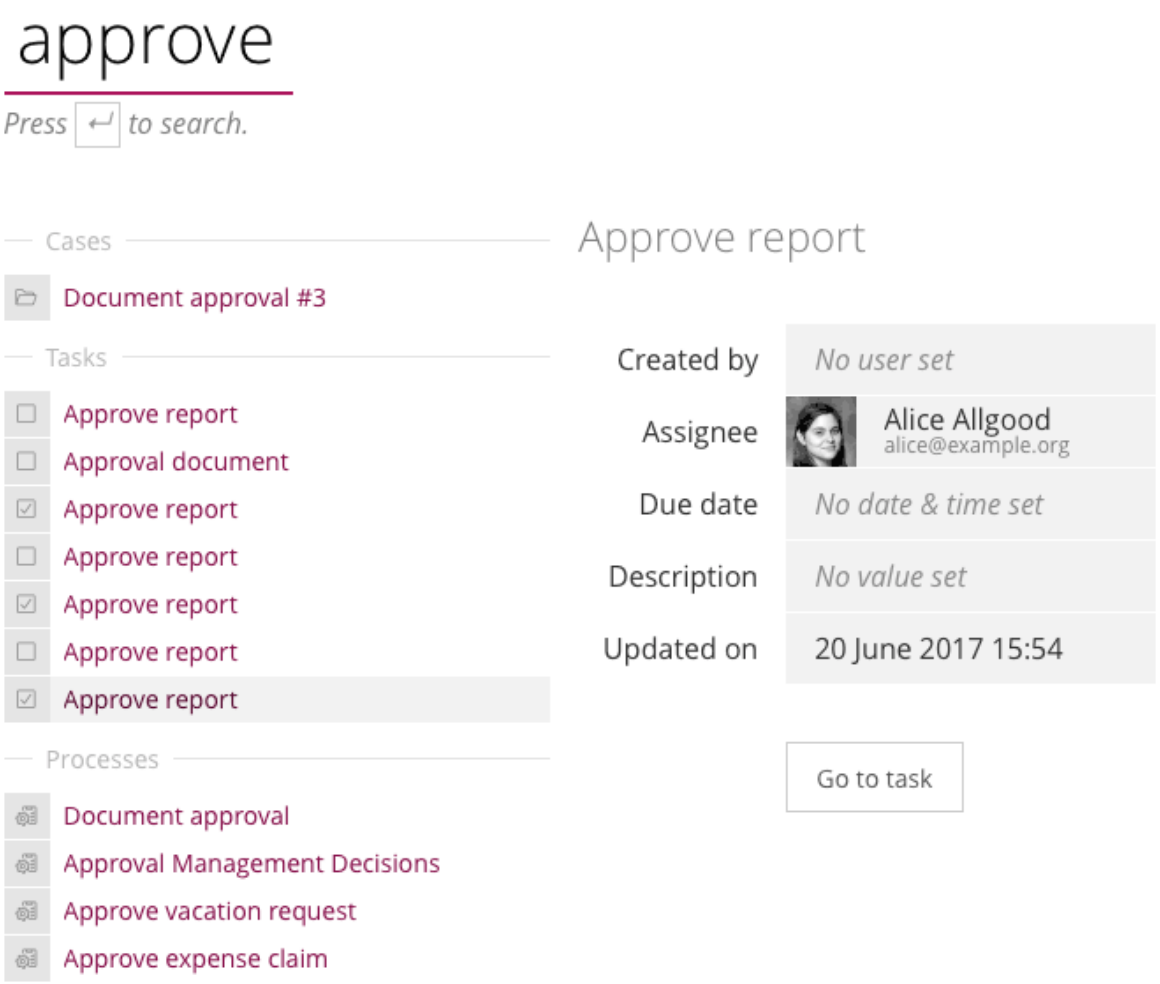
You can open the search page via the search option in the main menu bar.

Main menu search icon

Enter a search term.

Search page input field

The search results show a list with the first ten entries per category. If you do not find the required search result in the list, you need to refine your search by adding or excluding keywords, for example. Hover an entry to show details on the right.



Search results with one entry highlighted.

Note: The case icon next to the case name indicates the case status - open or closed.

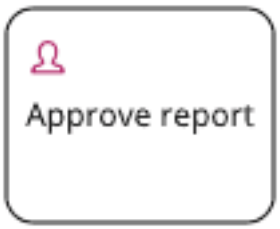
Select the case name to open the case.

Chapter 7

Action Types

The *process builder* (page 28) displays actions as rounded rectangles. Actions typically have a configuration panel that opens when you select the action.

7.1 User task



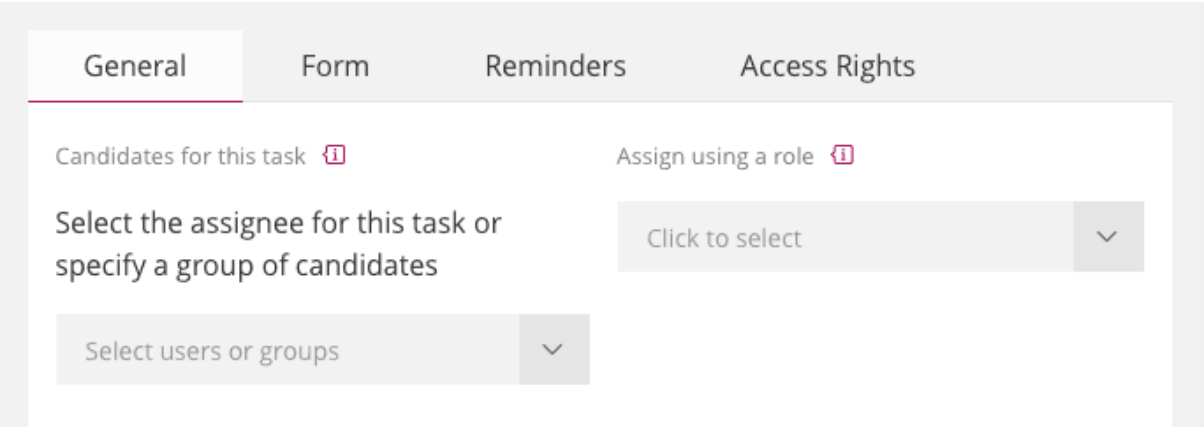
A user task in the process editor

A user task indicates that someone will perform a task. In the configuration form, you can specify the following optional details.

- *General* - assign a task to a user or a *process role* (page 43).
- *Form* - add a form to a task; submit the form to complete the task.
- *Reminders* - task deadline notifications.
- *Escalations* - automatic task reassignment.
- *Access rights* - task permissions for users and groups.
- *Auto-close task* - define a duration for the task to be completed, after which it will close automatically.

7.1.1 General

Use the *General* tab to specify the task's default assignee or candidates. You can select either individual users or organization groups as candidates.

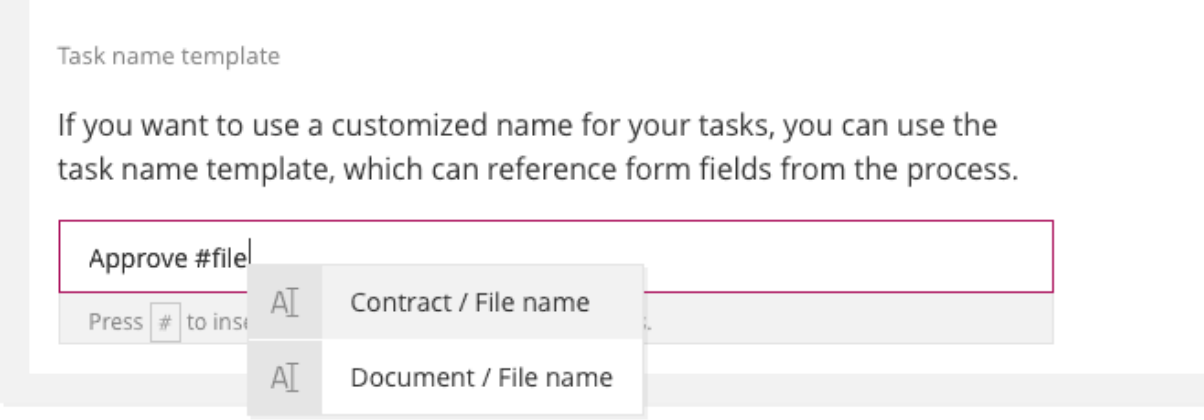


Configuring user task assignment in the process editor

You an also specify a role for the task's assignment, like a BPMN swim lane, so that Signavio Workflow Accelerator automatically assigns related tasks to the same person.

7.1.2 Task name template

You can use task name templates to create dynamic task names from *variables* (page 43). Enter the task name template on the user task's configuration panel's *General* tab. In the text field, type a # to select a variable.



Using a task name template to configure dynamic task names

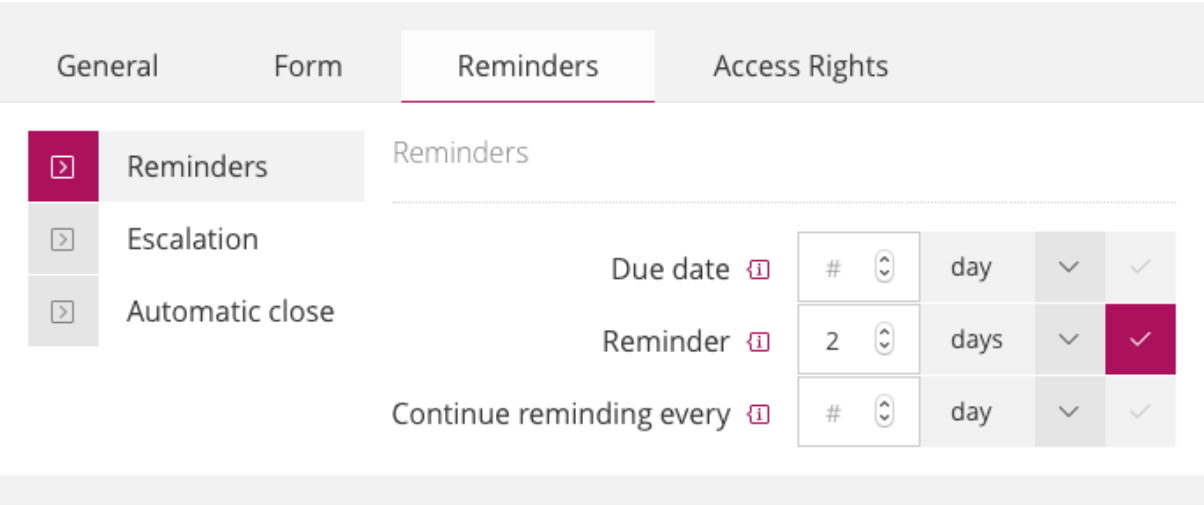
Use task name templates to avoid a *Tasks* (page 8) view that fills up with identically-named tasks.

7.1.3 Form

Use the form builder to create a user task form for entering and updating data as part of the user task.

7.1.4 Reminders

Use the user task's *Reminders* configuration panel tab to set up task deadline notifications. If you configure a *Due date* or *Reminder* period, then Workflow Accelerator will send email when the deadline expires.



Configuring user task reminders in the process editor

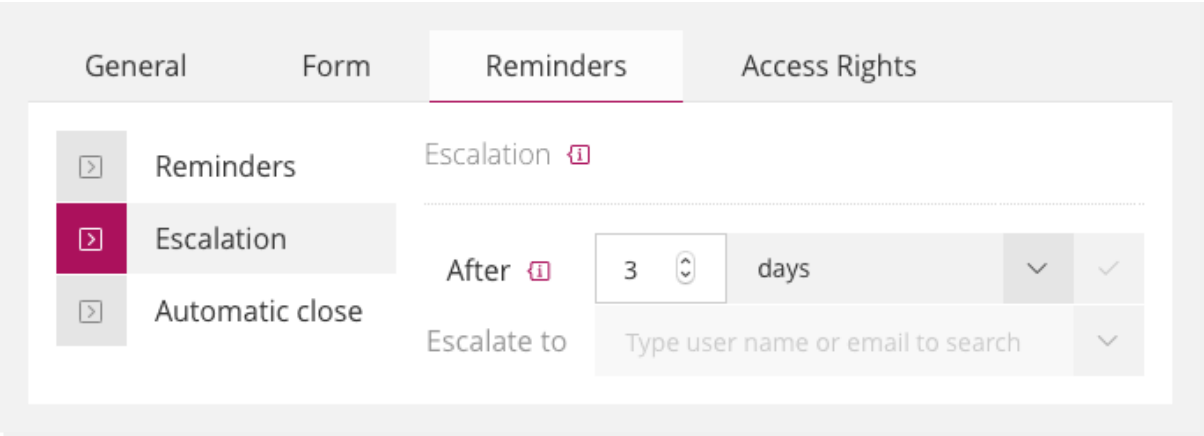
The three settings work as follows.

- *Due date* specifies an automatic *task due date* (page 11) relative to the task's creation date, which results in a *Case task due* (page 120) notification, and changes how the task appears in the *Inbox* (page 9)
- *Reminder* works separately from the due date and only sends a *reminder* (page 121) notification, which you can use to remind task assignees and candidates earlier than the task deadline
- *Continue reminding* further configures the *Reminder* by repeating the *reminder* (page 121) notification up to 25 times

Workflow Accelerator sends these notifications to the task's assignee if the task has an assignee, or to all of the task's candidates if the task it not assigned. If the task remains unassigned and does not have any candidates, Workflow Accelerator sends the notification to the process' owner.

7.1.5 Escalations

You can also use the *Reminders* (page 57) configuration panel tab to configure escalations. Use escalations to automatically reassign a user task when a deadline expires.



Configuring user task escalation in the process editor

1. On the *Reminders* (page 57) configuration panel, select *Escalations*.

2. In the *After* field, set a period to wait after the task creation date, at which point Workflow Accelerator will automatically reassign the task.
3. In the *Escalate to* field, select a user or candidate group to escalate to, who will receive a *task escalation* (page 121) notification.

Sometimes, you don't know who will take over a task that has passed its escalation deadline, so you choose one person in the process model. Instead, you can escalate task assignment to multiple people, or select a group.

7.1.6 Access rights

Use the *Access Rights* tab to set permissions for viewing, assigning and completing the task. See *Restricting access to user tasks* (page 108) for details.

7.1.7 Auto-close task

New in version 3.44.

This feature allows the setting of a specific duration of time to complete a task, after which it will automatically be closed. For example, you can use this feature to assign certain inspection tasks to managers, and give them a set time to complete them. If they don't complete them in the specified time frame, the task is closed automatically and the execution of your process continues.

Note: This feature currently does **not** work for tasks that require a manual decision. If you try to use it with a manual decision, Workflow Accelerator will stop executing your process at the gateway.

7.2 Multi-user task

New in version 3.28.

A multi-user task in the process editor

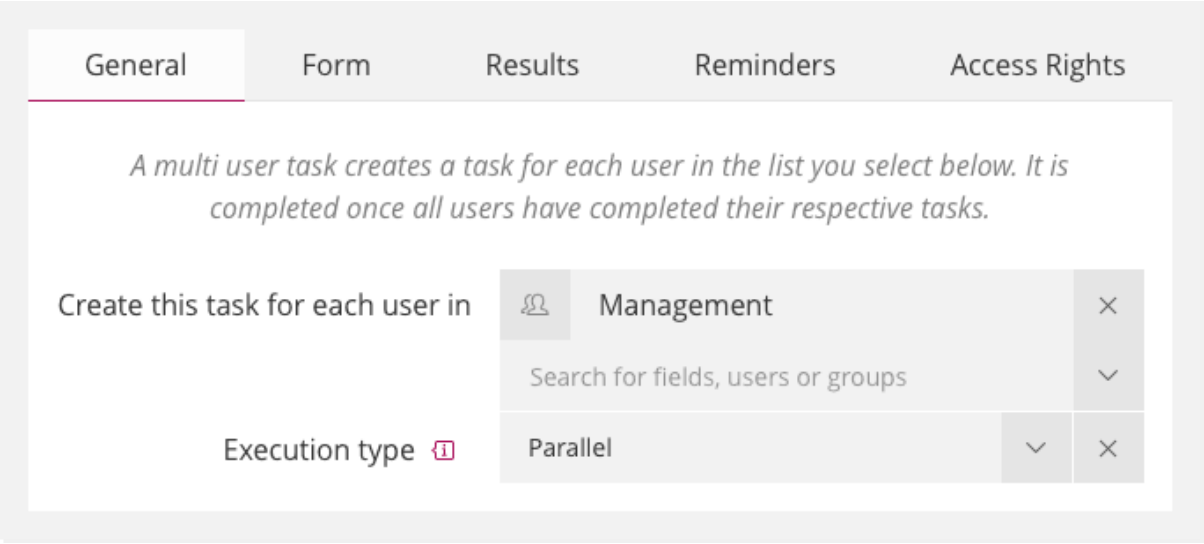
A multi-user task indicates that a group of people will each perform the same *User task* (page 56). You can use this to model multiple approvals, where several people in a group must approve a proposal.

In the configuration panel, you can set the same configuration as for a user task, plus the following additional configuration.

- *General* - assign tasks to users and groups
- *Results* - specify how form fields map to lists of values

7.2.1 General

Use the *General* tab to specify the users to create tasks for. The multi-user task creates a task for each user or group member.

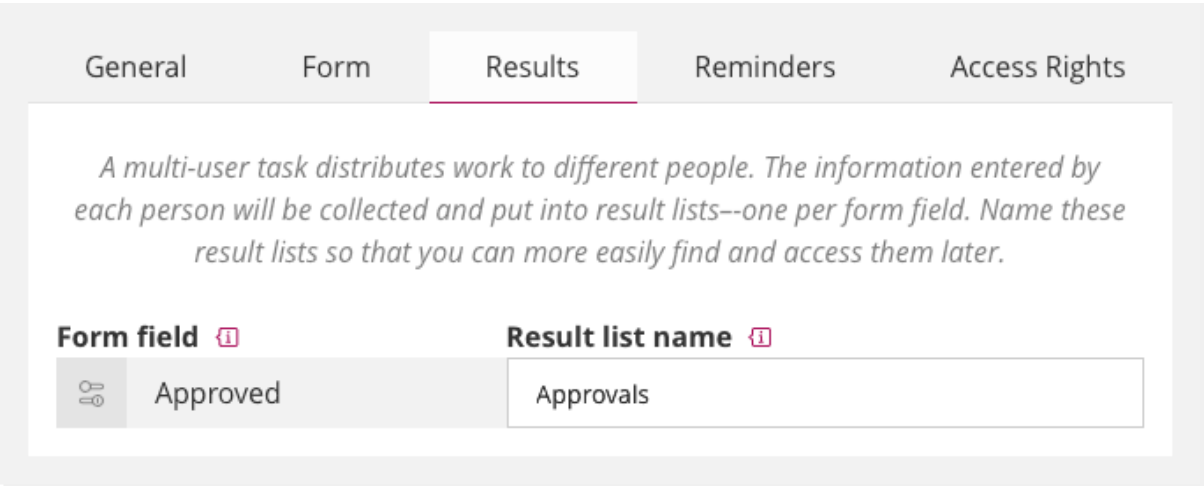


User and group configuration for a multi-user task

You can also specify parallel or sequential *Execution type*. If you select *Parallel*, the multi-user task will create all of the tasks at the same time, for their assignees to complete in any order. If you select *Sequential*, the multi-user task will create one task at a time, and wait for its assignee to complete it before creating another.

7.2.2 Results

Use the *Results* tab to map each form field to a list of values. This list collects the values entered by the people who complete the tasks that this multi-user task generates.



Form field result mapping for a multi-user task

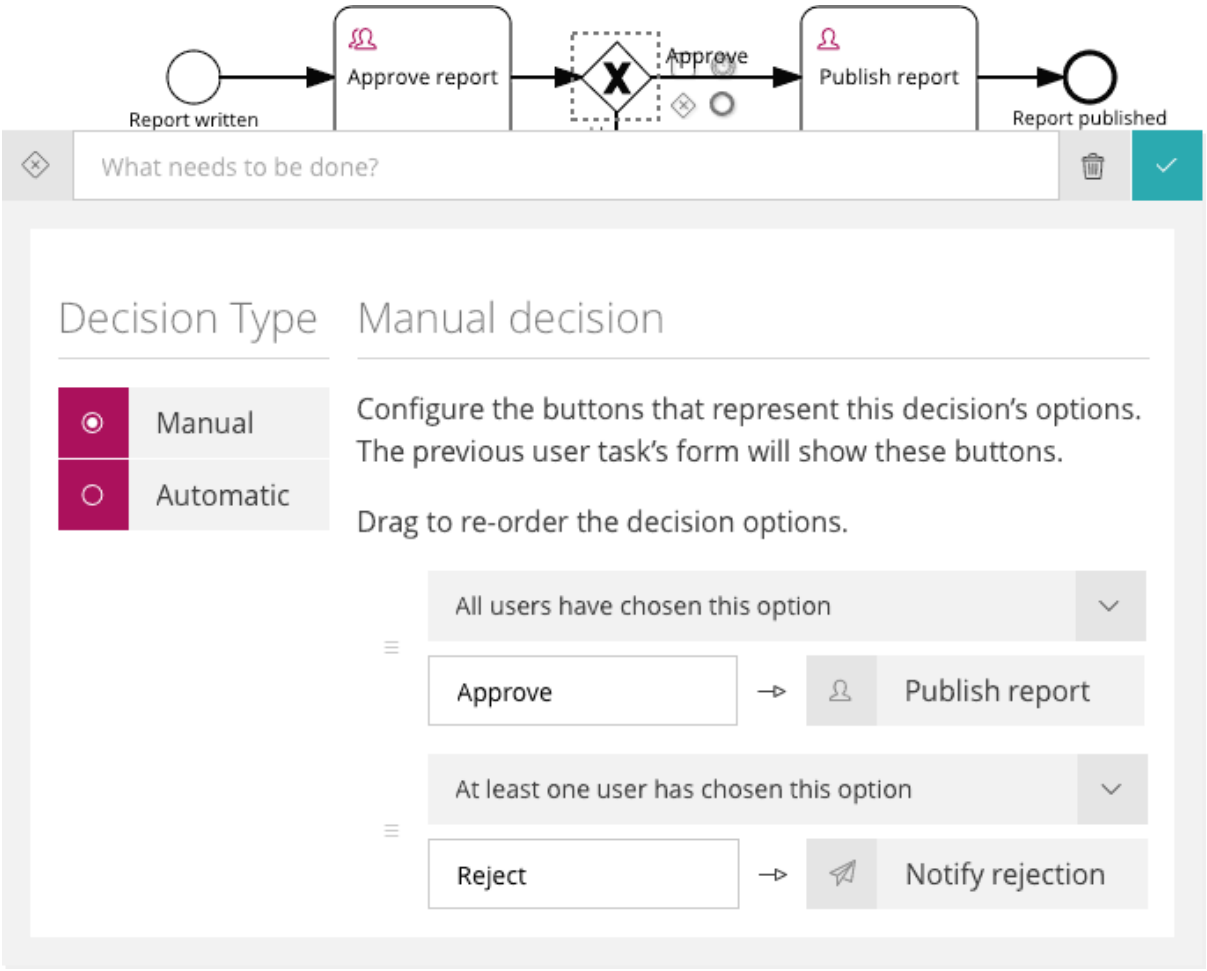
In this example, the multi-user task's form includes a single Yes/No checkbox field with the label *Approved*. The mapping to *Approvals* creates a variable with the same type as the *Approved* form field,

but which allows multiple values. Completing the tasks that the multi-user task generated adds the *Approved* values (Yes or No) to the *Approvals* list.

To use the results of a multiple approval, use a *JavaScript action* (page 74) to implement whatever logic consolidates the list of results in the list into a single decision.

7.2.3 Manual decision (Exclusive gateway)

The multi-user task supports manual decisions. If an exclusive gateway follows a multi-user task, you can configure the buttons that reflect the possible decisions to this gateway. The buttons are displayed in the form of the multi-user task.



Configuring decision options for a multi-user task

Select the decision type 'Manual' for the exclusive gateway. Then choose the decision option from the drop-down menu:

- *All users have chosen this option*
- *At least one user has chosen this option*

7.3 Send email

A send email task in the process editor

The send email action sends an email to the specified user. In the configuration you can specify the following.

Sender name By default, Workflow Accelerator sends the email with the *From* field set to *Workflow Accelerator*. You can set this field to set the *From* field to your company or department name.

To The people who Workflow Accelerator sends the email to. You can enter one or more plain email addresses, or select email *variables* (page 43) or users in the organization. Note that if you enter a plain email address, you must then select it from the pick list.

Reply to Workflow Accelerator sends emails from the address *notifications@workflow.signavio.com* (Europe server), *notifications@workflow-us.signavio.com* (US server), *notifications@workflow-au.signavio.com* (Australia server), which you cannot reply to. If you set this field to an alternative email address, you override the address for replies to the email.

Subject The email subject line. To use *variables* (page 43) in the subject, type a # and select a field from the list. If the list contains too many variables, you can just keep typing after the # to filter the list. You can use the arrow keys to move the section, and *Enter* to select a field. To remove a variable, just delete it the same way you delete normal text.

Attachments To add attachments, click the *Attachments* field and select a file field from the list. The list includes trigger email attachments and file upload form fields.

Body text You can use variables in the main email body by typing a #, just like in the *Subject*. You can use *Markdown* (page 129) to format the email body by formatting text or including hyperlinks, headings or lists. Select the *Preview* tab to see how Workflow Accelerator will format your email.

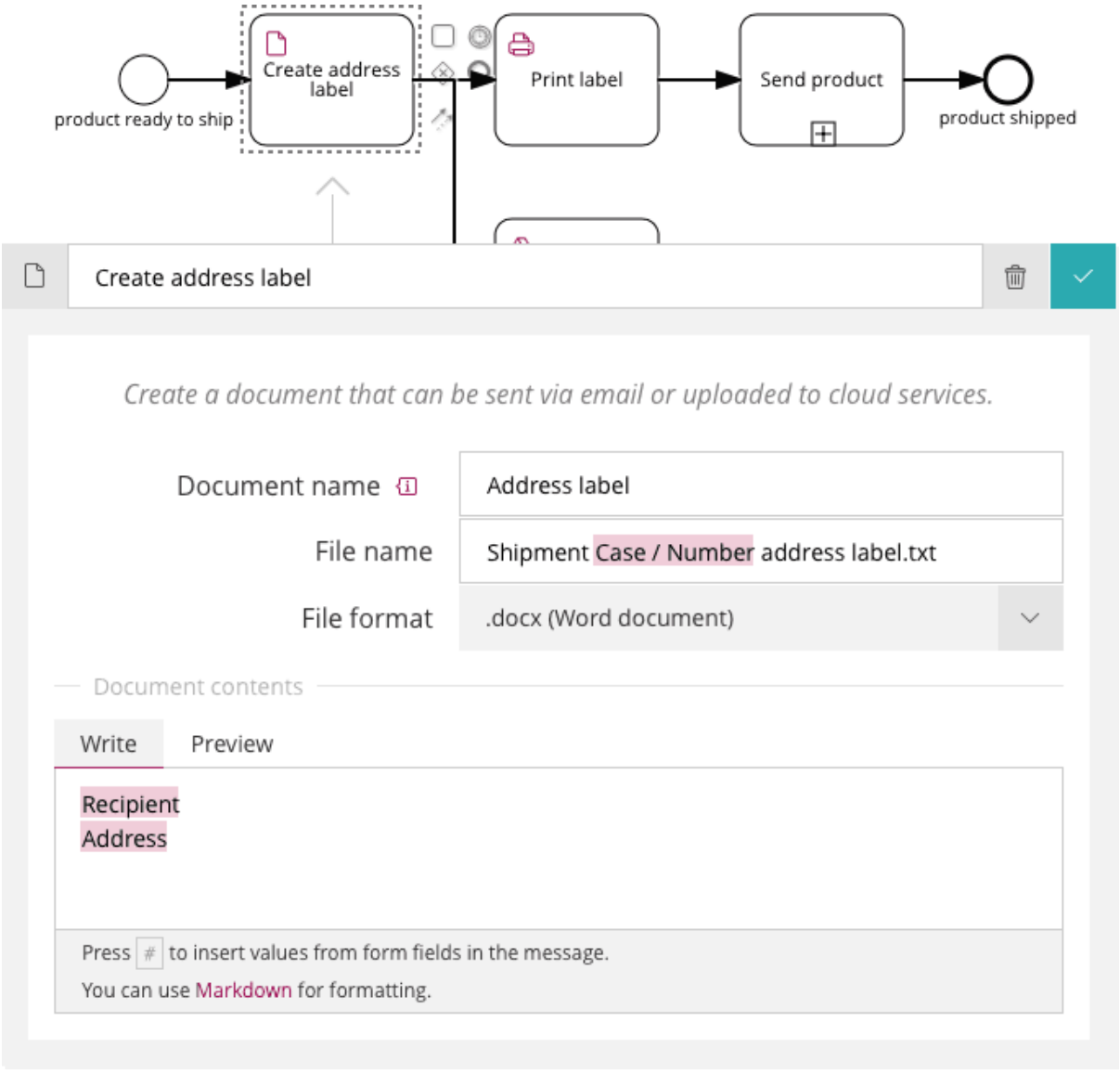
7.4 Create document

A create document action in the process editor

The “Create document” action allows you to create a file that contains case information.

When you use the *Send email action* (page 62) or *Upload file* (page 64) to save information from a case, you use a file variable for email attachments or the file to save. You normally provide these files via a form. You can also use the "Create document" action to create a new file using data from other variables.

Consider a shipping process whose trigger form includes details of a recipient to ship a product to and a shipping address. This process can use a "Create document" action to prepare a shipping label for printing:



Configuring a create document action in the process editor

Select a "Create document" action in the process editor to configure the following options.

Document name Defines the name of the file variable that will store the created document.

File name Defines the document's filename. Type # to include placeholders for variables.

File format Choose between Microsoft Word, plain text, and comma-separated values (CSV) file types.

Body text Use the text area to enter a template for the new document. Type # to include placeholders for variables, as you would in a *Send email action* (page 62)'s template.

Note that the "Create document" action does not currently support formatting text.

7.5 Google Drive - Upload file

A Google Drive Upload file *task in the process editor*

The Google Drive “Upload file” action sends one or more files to an account of your choice.

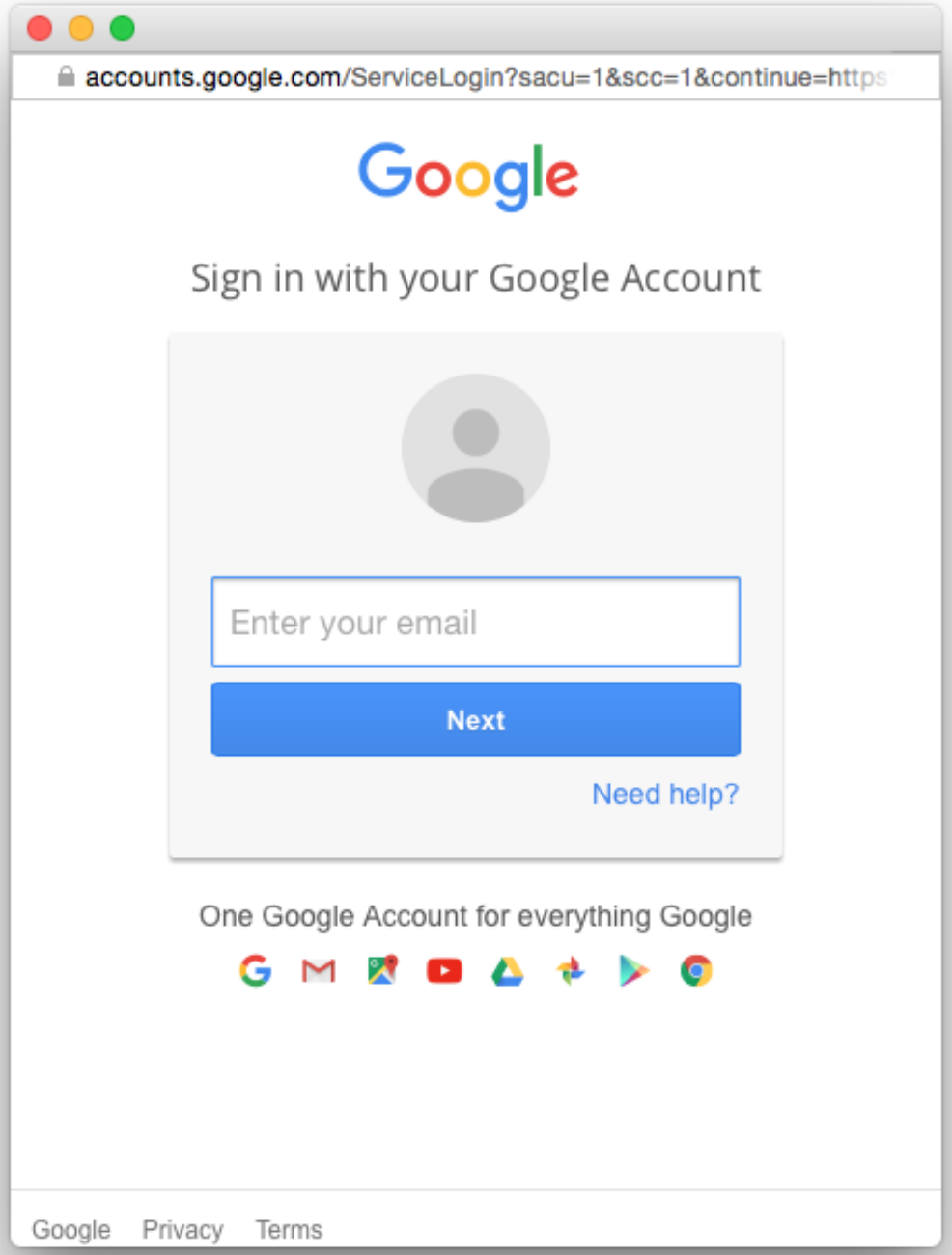
7.5.1 Configuration

After creating a Google Drive “Upload file” action, the configuration panel shows a button to start configuring the account.

When you click on the button “Configure a Google Drive account”, a pop-up window will appear that helps you integrate Workflow Accelerator with your Google Drive account.

Google will check that you have already logged in to Google. This check has three possible outcomes:

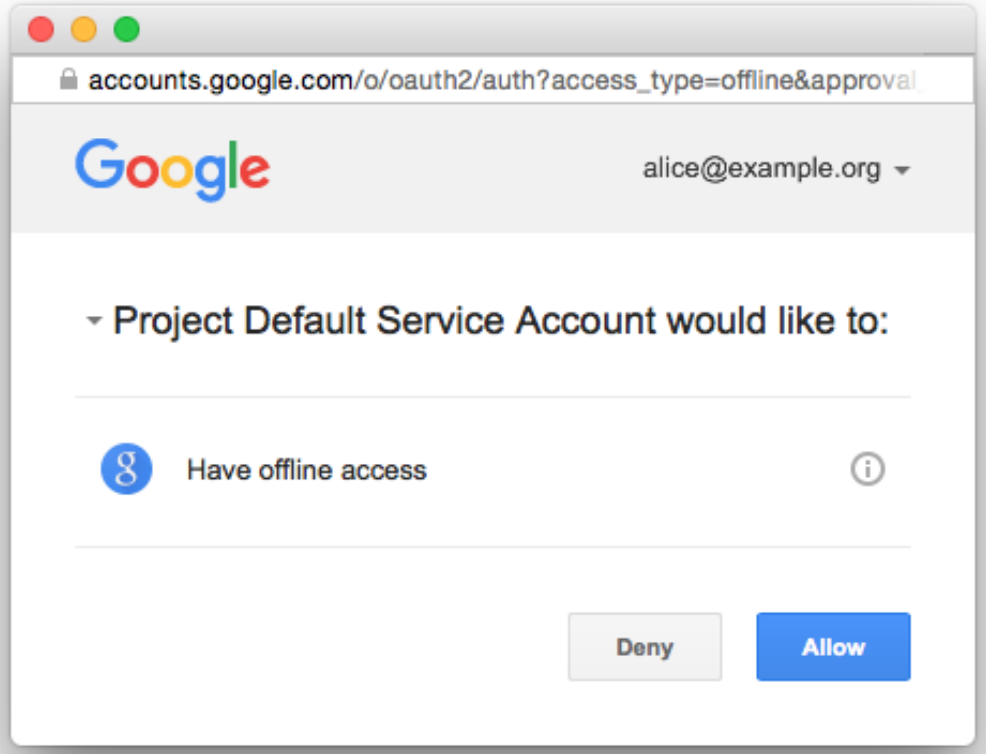
1. If you have already authenticated with a single user to Google, the set-up process skips the log in page and you go straight to the *permission screen* (page 66) below.
2. If you haven’t authenticated with Google, you will see a log in window:



Google Drive login

3. If you have authenticated with multiple Google accounts, select the account you want to use.

After you have completed authentication, you'll see the following page that allows you to give Workflow Accelerator access to your Google account.

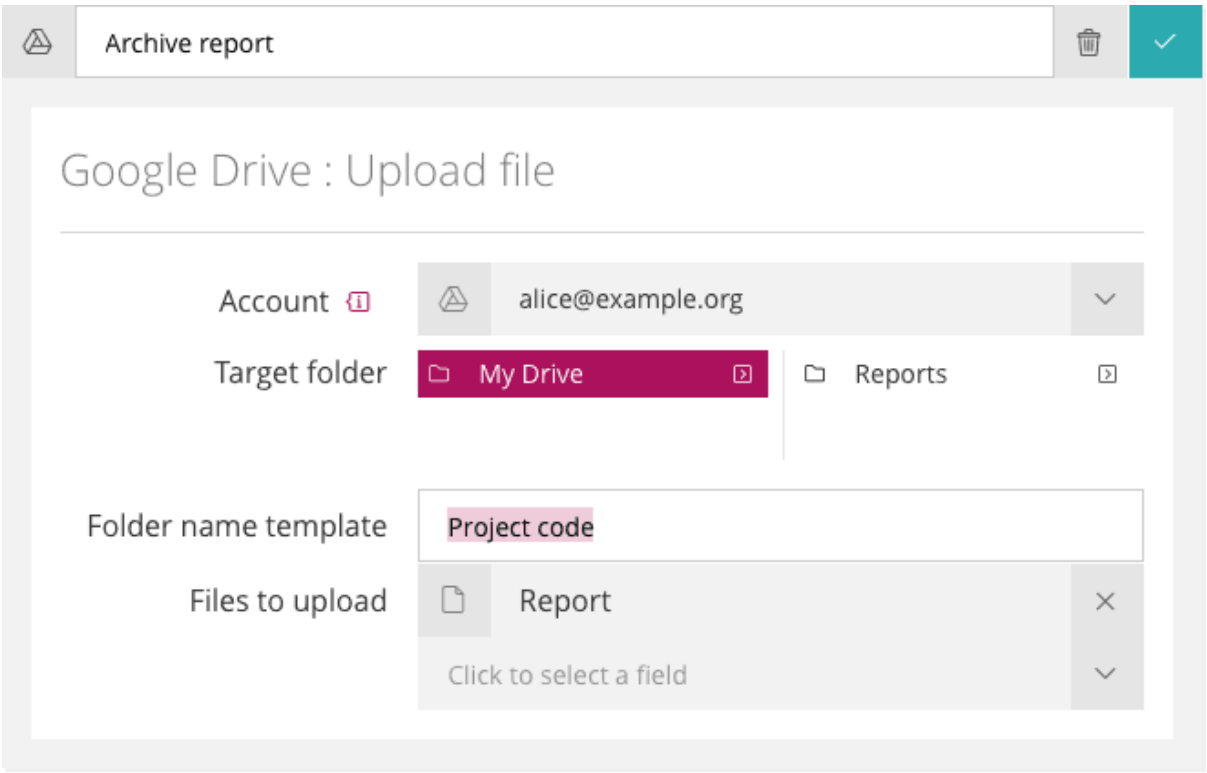


Google Drive permission grant

After accepting the permissions, the pop up will disappear and Workflow Accelerator will have completed connecting to your account.

7.5.2 Upload file action

Once you have configured your Google Drive account, the *Upload file* action configuration panel displays the account and also shows the folders in “My Drive” in your Google Drive account.



Google Drive Upload file *authenticated configuration*

Other people in your organization can see that you have configured an account, but they cannot see your email address or browse your account folders.

In the "Target folder" section you can now browse and select the folder you want to upload the file(s) to.

Next to "Folder name template", you can optionally specify a subfolder name to create inside the target folder. This name template can include process variables, so you can create new subfolders dynamically to organize your files. For example, if your process variables include a unique customer ID, then you can use that to save each customer's files in a separate folder. Use a forward slash (/) to separate nested subfolder levels.

Next to "Files to upload", you can select the variable field containing one or more files to upload. If the process did not already include a file variable, Workflow Accelerator automatically creates a variable called 'File'. If the process variables did include a file or list of files, Workflow Accelerator preselects it.

7.6 Google Drive - Print file

A Google Drive Print file *task in the process editor*

The Google Drive “Print file” action uses Google Cloud Print[12] to print a file that a case participant has uploaded to a case.

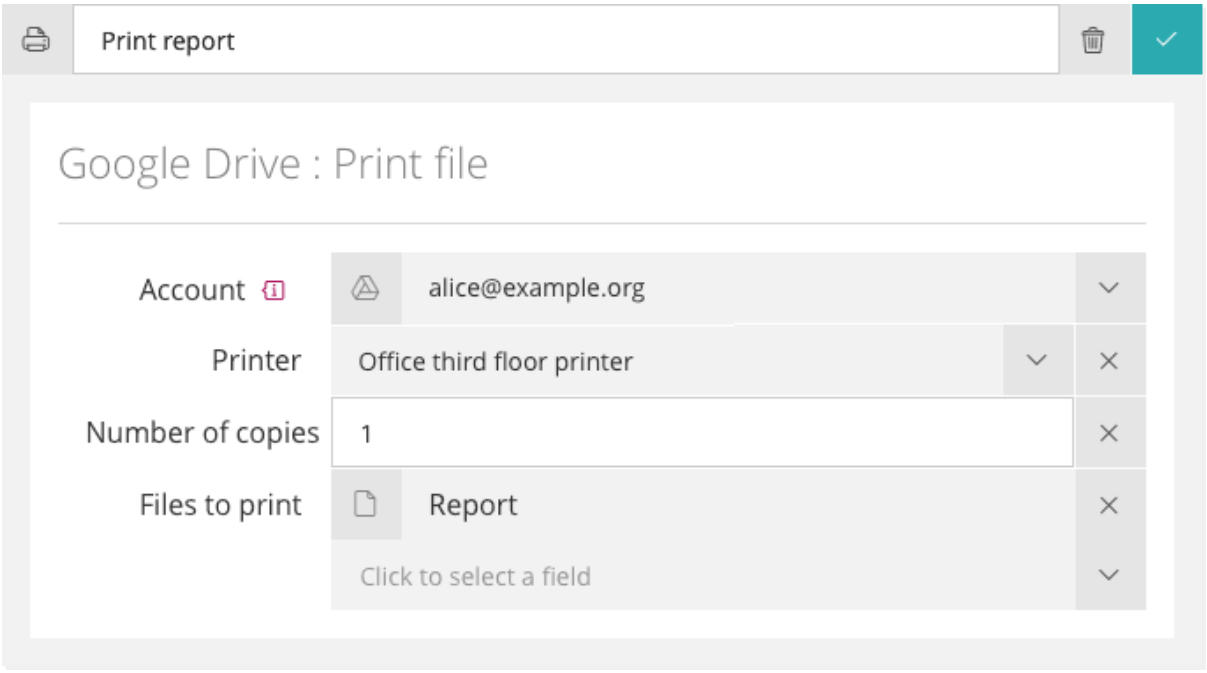
7.6.1 Configuration

After creating a Google Drive “Print file” action, the configuration panel shows a button to start configuring the account. This configuration has the same steps as in the *Upload file configuration* (page 64) (above).

When you have authenticated with a Google account and granted permission, you can configure the “Print file” action.

7.6.2 Print file action

Once you have configured your Google Drive account, the “Print file” action configuration panel displays the account, the printer settings, and the selected file to print.



Google Drive “Print file” authenticated configuration

In the section “Files to upload” you can select the file variable that holds the file you want to print.

[12] http://www.google.com/cloudprint/learn/

7.7 Google Drive - Add row to sheet

A Google Drive Add row to sheet *task in the process editor*

The Google Drive *Add row to sheet* action adds a row to a Google Sheets spreadsheet. You can use this to save the values of process variables at process milestones, and build a custom overview of cases.

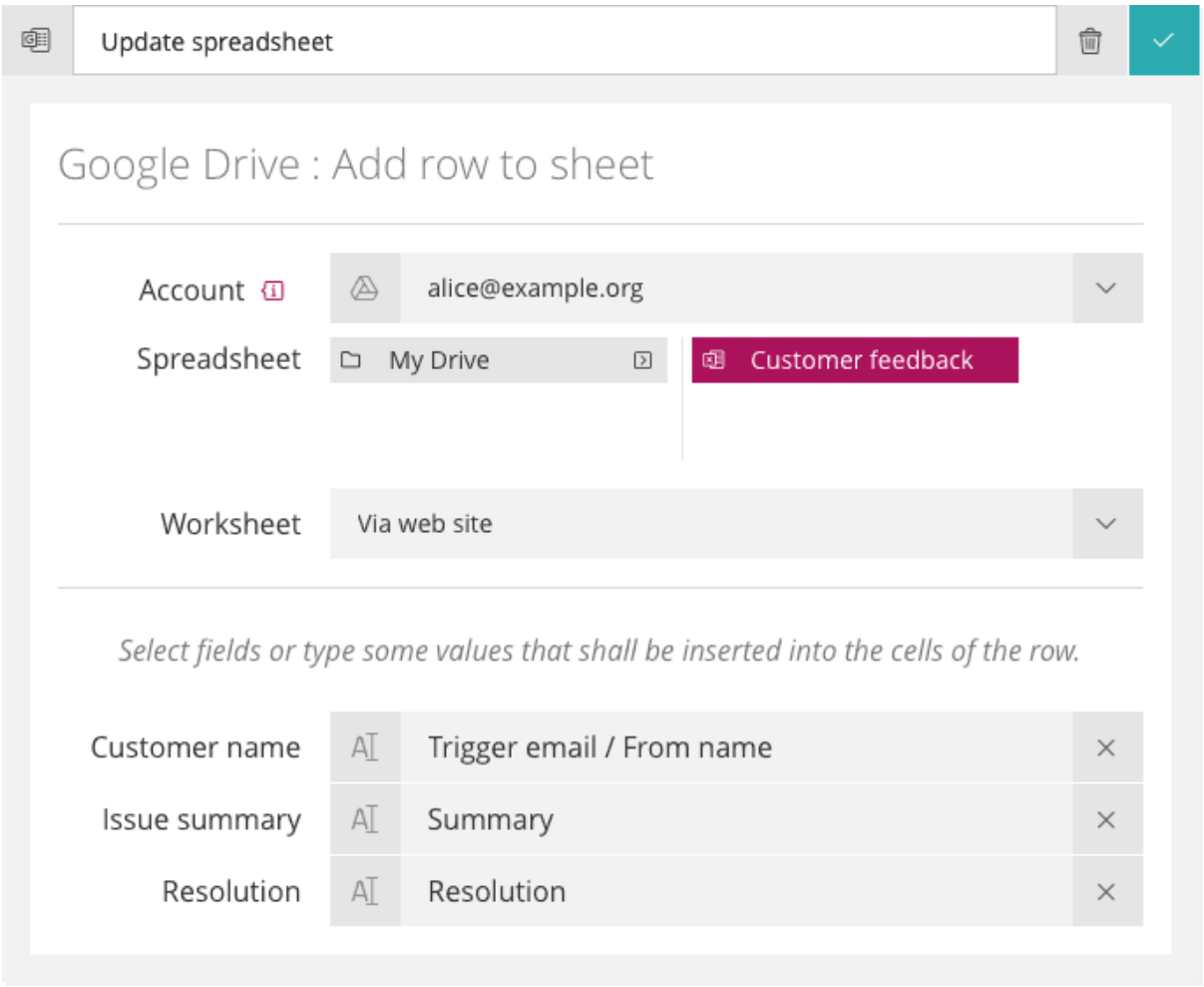
7.7.1 Configuration

After creating a Google Drive *Add row to sheet* action, the configuration panel shows a button to start configuring the account. This configuration has the same steps as in the *Upload file configuration* (page 64) (above).

When you have authenticated with a Google account and granted permission, you can configure the *Add row to sheet* action.

7.7.2 Add row to sheet action

Once you have configured your Google Drive account, the *Add row to sheet* action configuration panel displays the account, the spreadsheet, the worksheet within the spreadsheet, and the worksheet columns.



Google Drive Add row to sheet *authenticated configuration*

Each column name, such as *Customer name* in this example, comes from a column heading in the spreadsheet. For each column, select one of the *variables* (page 43) from the list.

7.8 Google Drive - Add calendar event

A Google Drive Add calendar event *task in the process editor*

The Google Drive *Add calendar event* action adds an event to a Google Calendar. You can use this to schedule meetings or time to work on a task, based on the values of process variables.

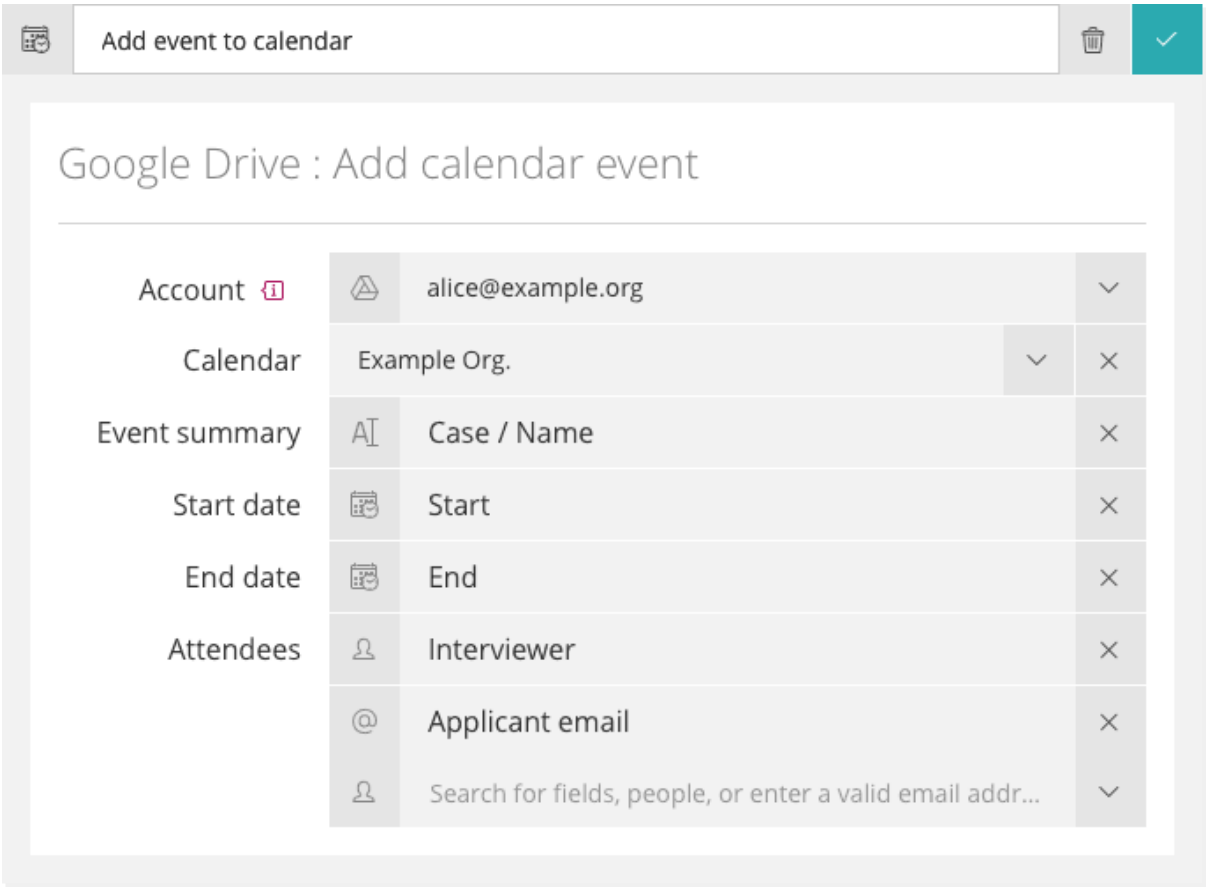
7.8.1 Configuration

After creating a Google Drive *Add calendar event* action, the configuration panel shows a button to start configuring the account. This configuration has the same steps as in the *Upload file configuration* (page 64) (above).

When you have authenticated with a Google account and granted permission, you can configure the *Add calendar event* action.

7.8.2 Add calendar event action

Once you have configured your Google Drive account, the *Add calendar event* action configuration panel shows the calendar event fields.



Google Drive Add calendar event *authenticated configuration*

In the configuration you can specify the following.

Calendar The calendar within the selected Google account.

Event summary (optional) A text variable to use as the name of the new calendar event. If you don't select a text variable, the event will have a blank name.

Start date A date variable for the event's start date and time.

End date A date variable for the event's end date and time.

Attendees (optional) Email address variables for people to invite to the calendar event.

7.9 Box Upload file

A box upload file task in the process editor

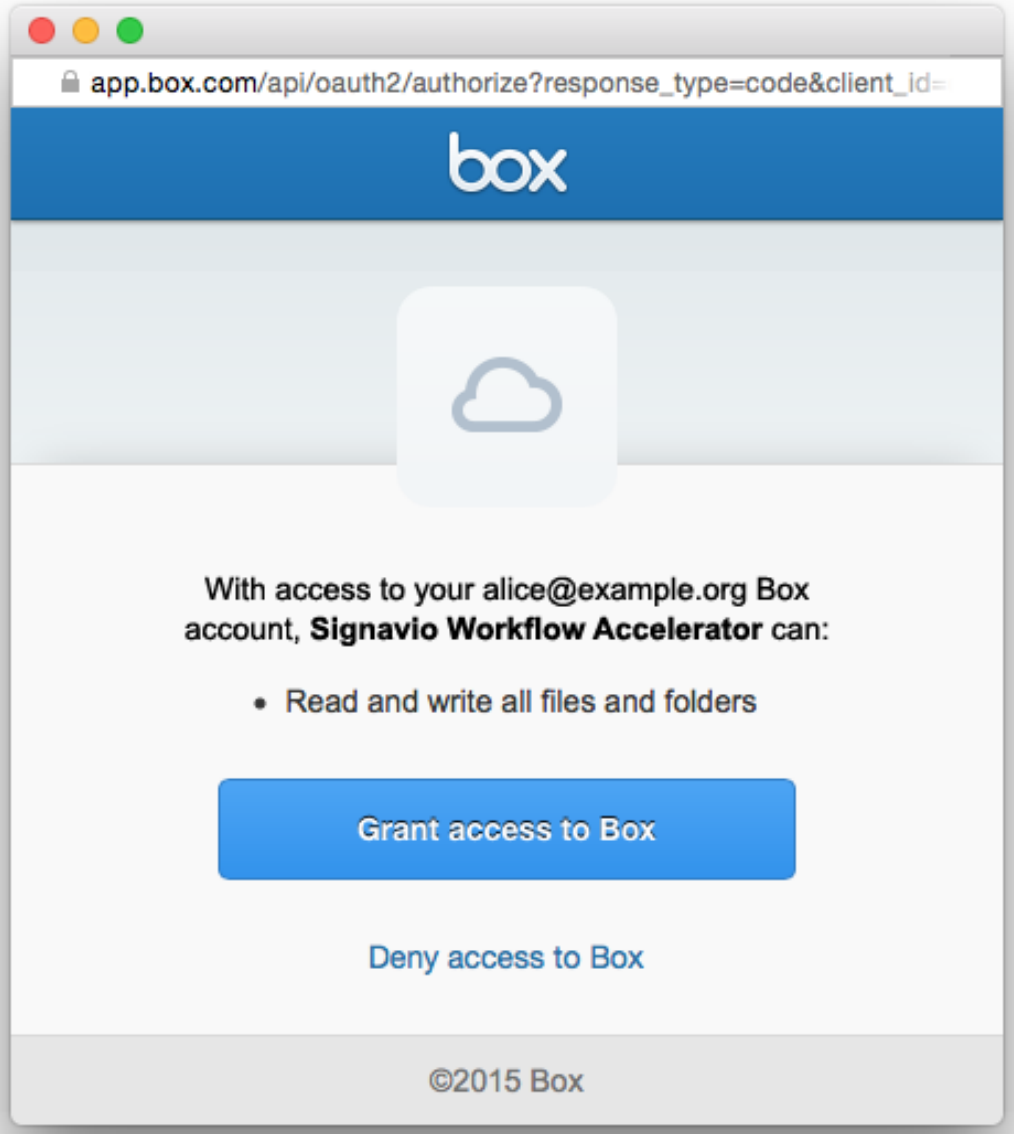
The Box[13] *Upload file* action saves one or more files to a Box account that you select.

7.9.1 Configuration

After creating a Box "Upload file" action, the configuration panel shows a button to start configuring the account.

When you click the "Configure a Box account" button, a pop-up window will open for you to authorize Workflow Accelerator to use your Box account. After logging in to your Box account, if you have not already logged in, Box shows an authorization page.

[13] http://www.box.com/

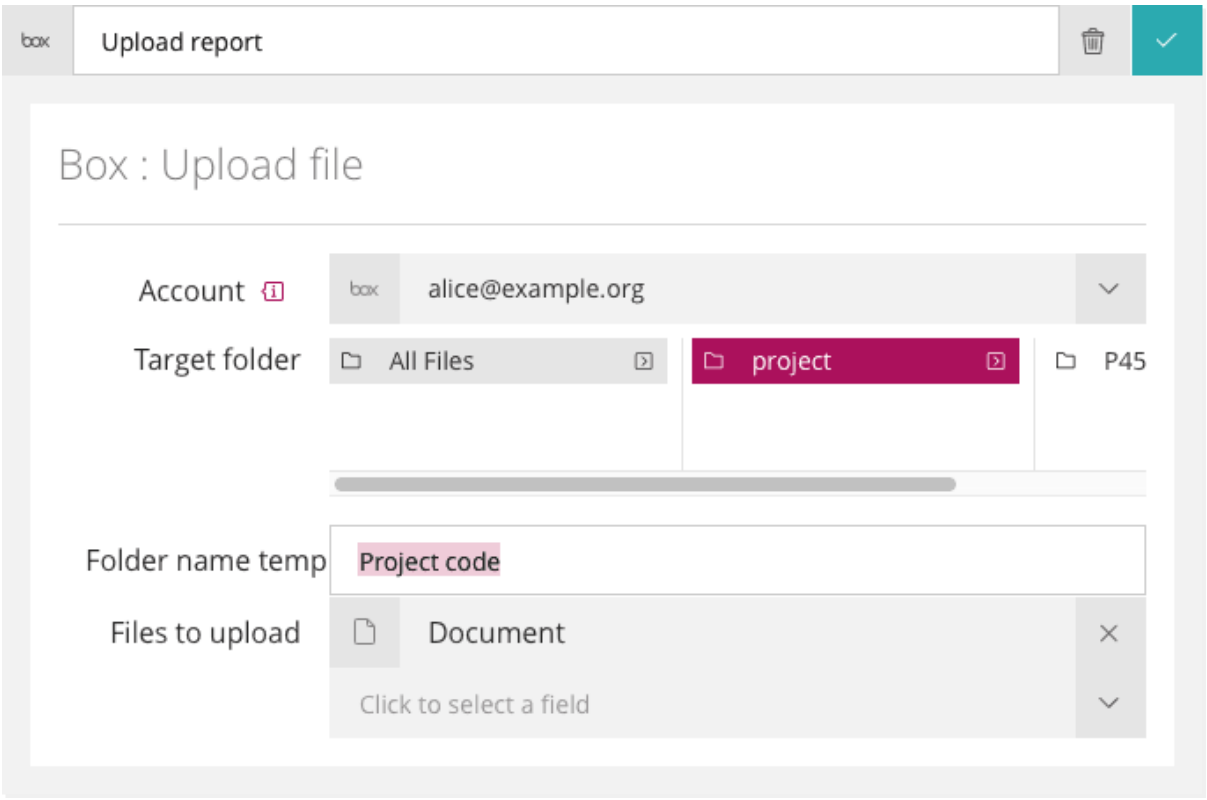


Box permission grant, after logging in to Box

After granting access, the pop up will close and Workflow Accelerator will have connected to your Box account.

7.9.2 Upload file action

Once you have configured your Box account, the “Upload file” action configuration panel displays the account and also shows the folders in “All Files” in your Box account.



Box authenticated configuration

Other people in your organization can see that you have configured an account, but they cannot see your email address or browse your account folders.

On the configuration panel, next to “Target folder”, you can now browse and select the folder you want to upload the file(s) to.

Next to “Folder name template”, you can optionally specify a subfolder name to create inside the target folder. This name template can include process variables, so you can create new subfolders dynamically to organize your files. For example, if your process variables include a unique customer ID, then you can use that to save each customer’s files in a separate folder. Use a forward slash (/) to separate nested subfolder levels.

Next to “Files to upload”, you can select the variable field containing one or more files to upload. If the process did not already include a file variable, Workflow Accelerator automatically creates a variable called ‘File’. If the process variables did include a file or list of files, Workflow Accelerator preselects it.

7.10 JavaScript action

The JavaScript action, called a *Script Task* in BPMN, allows developers to add JavaScript code to process execution.

A JavaScript action in the process editor

See *JavaScript Integration* (page 134) for a developer guide to using JavaScript actions.

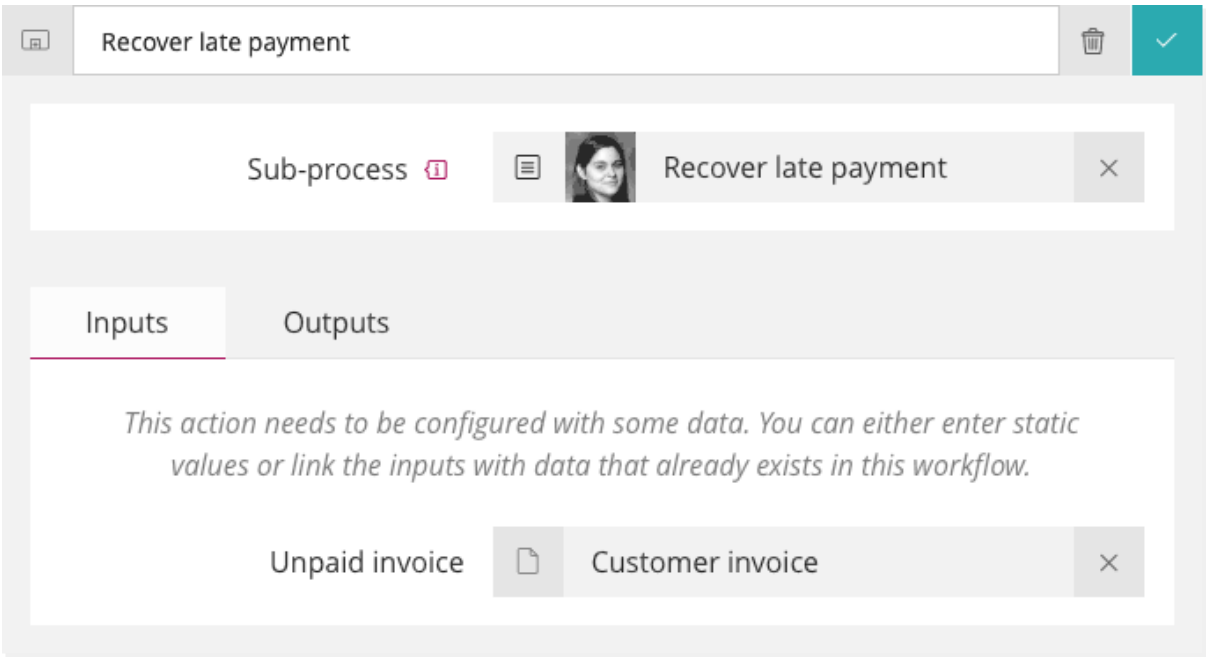
7.11 Sub-process

A sub-process in the process editor

A sub-process action represents an entire process as a single task. You can use sub-processes to simplify complex process models, by collapsing each sub-process into a single action. Then you can open the sub-process to see the next level of detail. You can also use a sub-process to delegate responsibility for the process model to a different process owner.

When Workflow Accelerator creates a sub-process, it starts a new case in the selected sub-process. After completing the sub-process case, Workflow Accelerator continues executing the parent case.

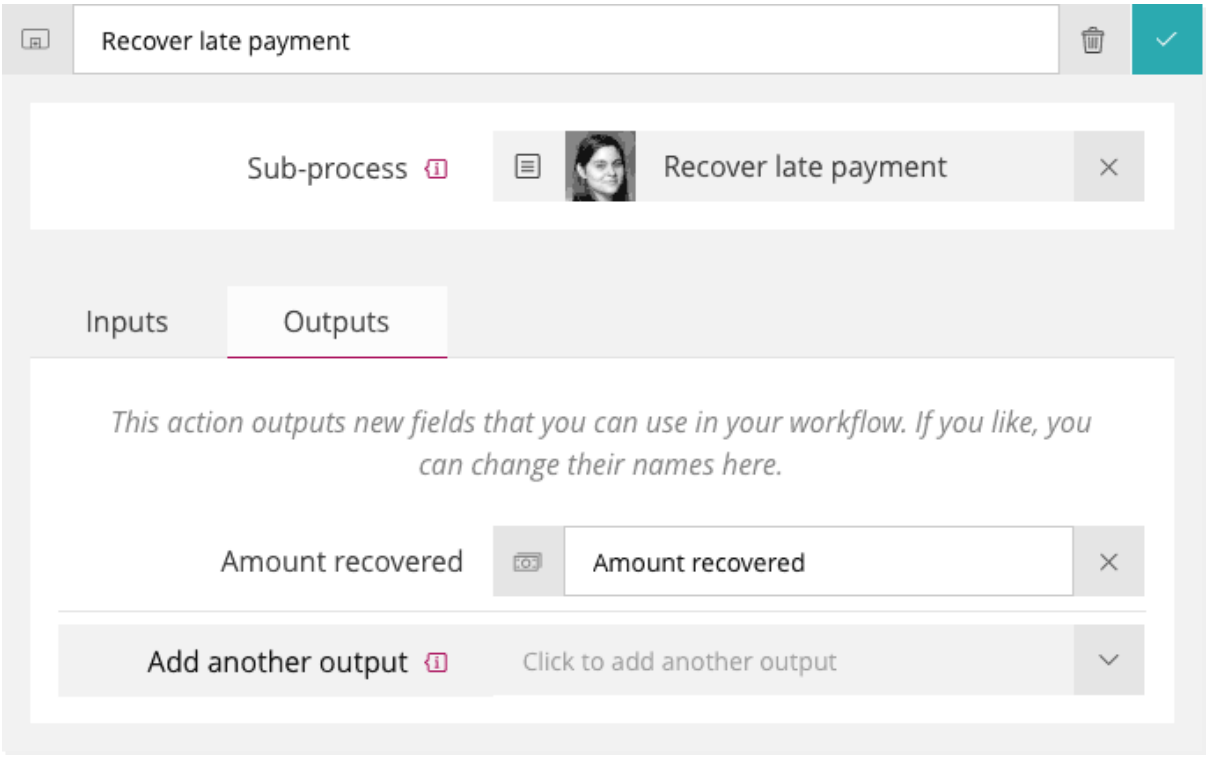
Configure the sub-process action in the process builder by selecting it, and choosing a process from the list. You cannot select the same process as the parent process, which would cause a loop, or an unpublished process.



Configuring a sub-process and mapping an Invoice *variable to a trigger form field*

If you select a sub-process with a trigger form, you can also select 'input variables' that Workflow Accelerator will use to auto-fill the trigger form fields when starting the sub-process case. In the example above, Workflow Accelerator will populate the sub-process' *Unpaid invoice* trigger form field with the file stored in the parent process' *Invoice* variable.

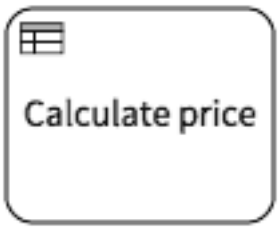
If you want to capture the result of running a sub-process, you can use the *Outputs* tab to add 'outputs'. Adding an output creates a variable that gets its value from a field in the sub-process when the sub-process completes.



Adding an output variable that gets its value from a sub-process field

In this example, the *Recover late payment* process has an *Amount recovered* variable. You can capture the value of this variable when the sub-process completes and use it later, in the parent process' *Check amount recovered* task.

7.12 DMN Rule Task



A DMN Rule task in the Workflow Accelerator process builder

DMN Rule Tasks execute Decision Model & Notation (DMN) business rules as part of a process. You can use rule tasks to embed complex business logic in a workflow, without adding complexity to the process model.

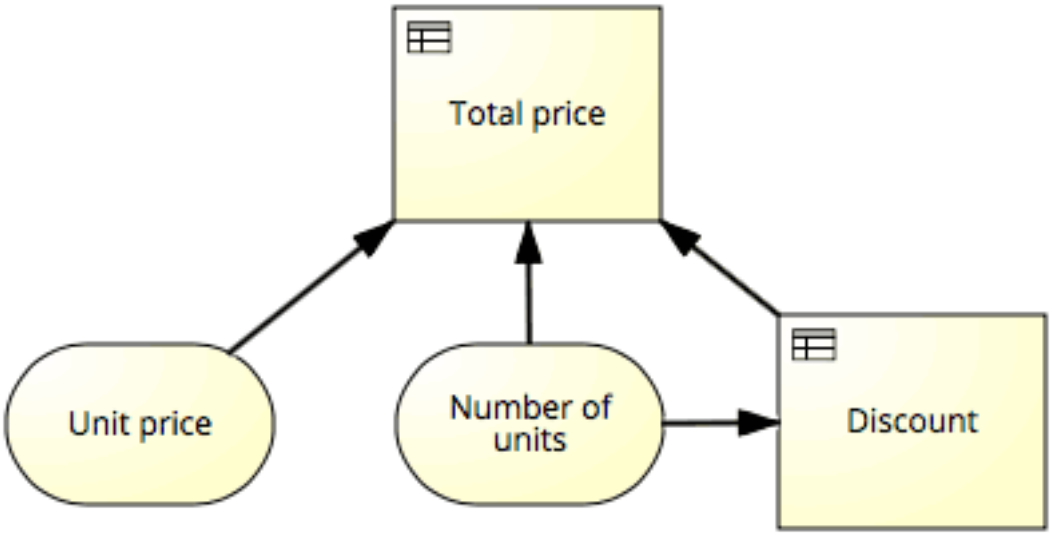
Note: To use DMN Rule Tasks, you need to have access to Signavio Process Manager[14].

Adding a DMN Rule Task to a process model requires the following steps.

1. Use Signavio Process Manager to create a DMN model.
2. Use Signavio Process Manager to create a BPMN process model.
3. In Signavio Process Manager, link a rule task to the DMN model.
4. In Signavio Explorer, transfer the BPMN model to Workflow Accelerator.
5. Use Workflow Accelerator to configure the rule task's inputs and outputs.

A simple price calculation example illustrates how to use DMN with Workflow Accelerator. Suppose you have a price quotation process that involves and order line price calculation, and that the price depends on a discount based on the number of items. Model this calculation in Signavio Process Manager by creating the following *Calculate price* model.

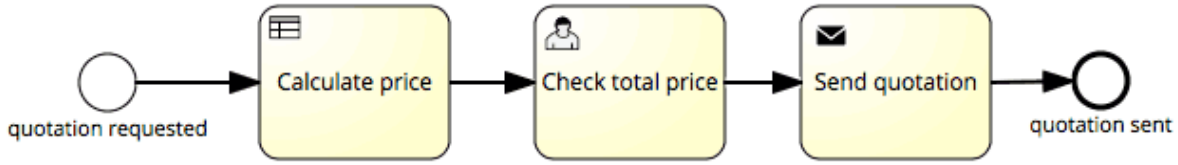
[14] http://www.signavio.com/products/process-manager/



A DMN model in Signavio Process Manager

This decision model has two numeric inputs, *Unit price* and *Number of units*. The *Discount* rule uses the *Number of units* to calculate a discount. The model has one output, *Total price*, which it calculates from the two inputs and the discount.

Next, model the price quotation process in Signavio Process Manager, by creating the following model. Link the *Calculate price* rule task to the *Calculate price* DMN model that you created earlier.



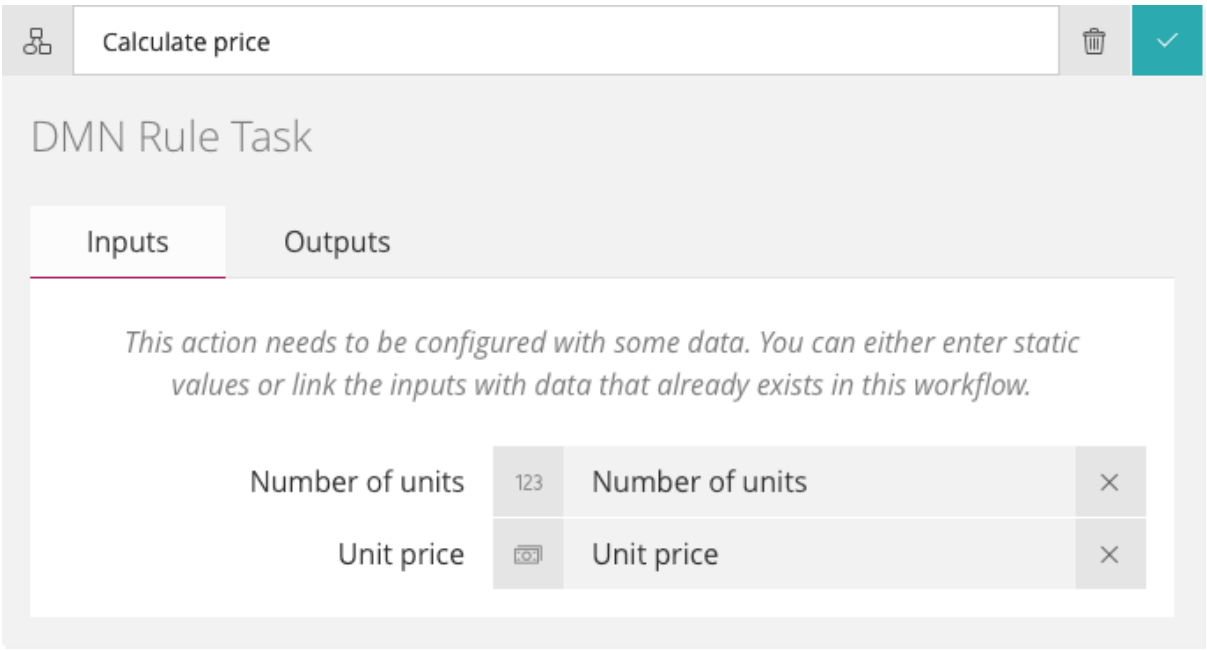
A BPMN process model, with a Calculate price *rule task, in Signavio Process Manager*

This process model starts with a *Calculate price* rule task, which calculates a total price, followed by a user task to check the result and an send task to send the quotation. Now save the model, so you can transfer it to Workflow Accelerator.

Next, select the BPMN model in the Signavio Explorer, and on the *Workflow Accelerator* menu, select *Execute in Workflow Accelerator*. On the dialogue box that opens, select *Transfer*, followed by *Open the workflow in Workflow Accelerator*.

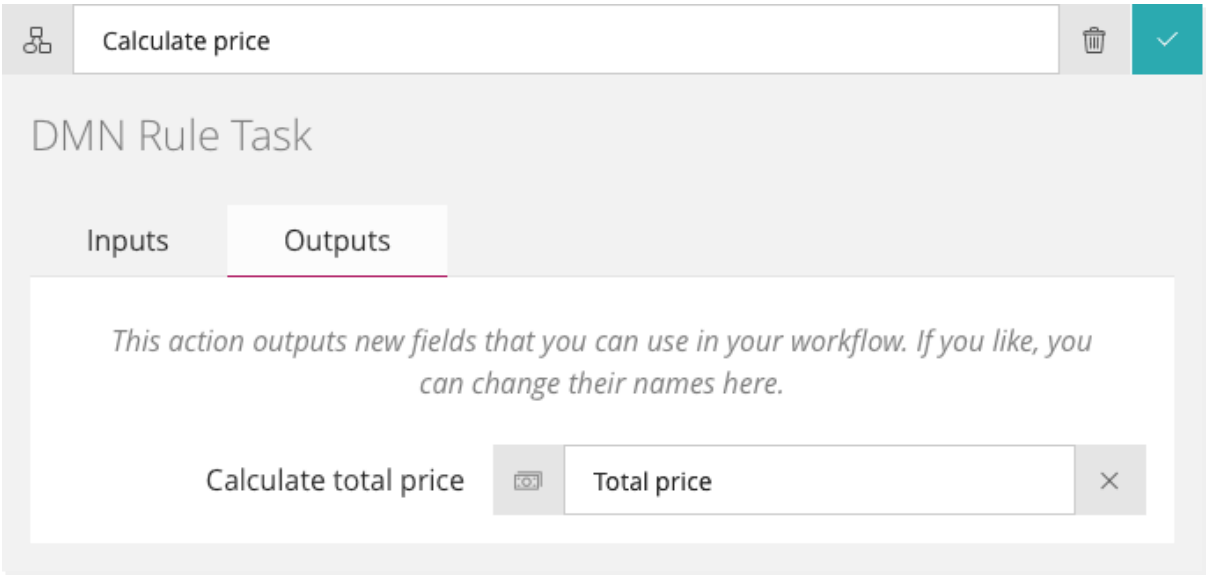
In Workflow Accelerator, you now need to edit the process to make it ready for execution. In the process builder, select *Trigger* and add a trigger form. On the trigger form, add two *Number* fields and name them *Unit price* and *Number of units*.

Select *Actions* and then select the *Calculate price* rule task to open its configuration panel. The *Inputs* shows the DMN model's two inputs. For each input, select the process variable of the same name, that you added to the trigger form.



Configuring a rule task's inputs in Workflow Accelerator

Select the *Outputs* tab to configure the output variable the rule task creates. The output variable has the default name *Total price* from the DMN model.



Configuring a rule task's outputs in Workflow Accelerator

You can now use the calculated *Total price* variable else where in the process. Add the existing *Total price* field as a read-only field to the *Check total price* user task. Finally, add the *Total price* field to the *Send quotation* email task's template, to send the quotation.

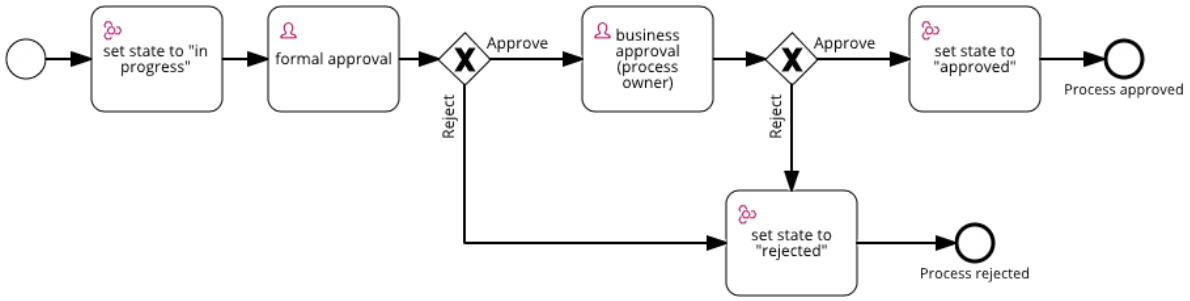
7.13 Signavio - Set model state

A Set model state *action in the Workflow Accelerator process builder*

When you use Signavio Process Manager[15] for process modelling, you can use Workflow Accelerator to manage process diagram approvals. Signavio Process Manager triggers these process diagram approval workflows, which run in Workflow Accelerator and in turn update the process model in Signavio Process Manager.

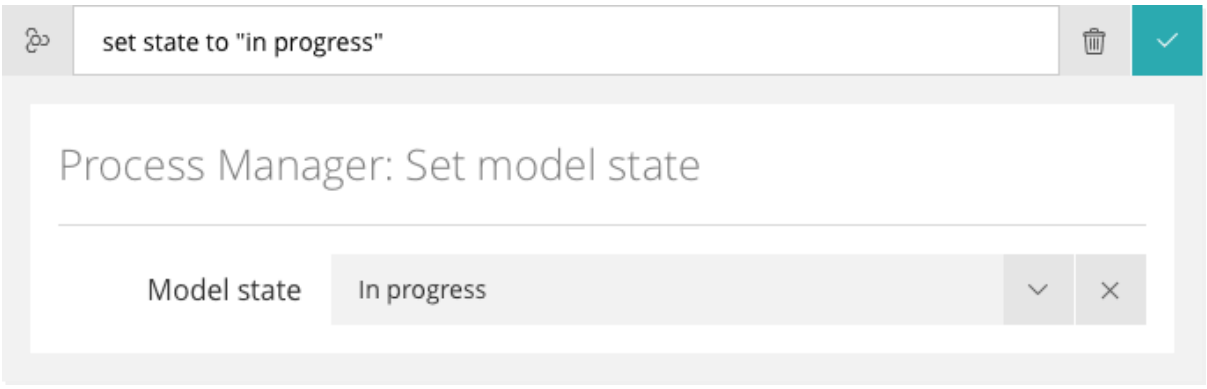
Note: To use the *Set model state* task, you need to have access to Signavio Process Manager[16]. This task only works in a process that has the *Signavio approval* trigger.

The *Set model state* action automatically updates the 'diagram state' in Signavio Process Manager, e.g. to mark the diagram as *approved* or *in progress*. You typically use this as part of a process that performs a management approval that marks the diagram as *accepted* or *rejected*.



A process diagram approval workflow that uses *Set model state* actions

Select a *Set model state* action to configure which process *Model state* the *Set model state* action will set.



[15] http://www.signavio.com/products/process-manager/
[16] http://www.signavio.com/products/process-manager/

Configuring a *Set model state* action to set a Signavio Process Manager diagram's status

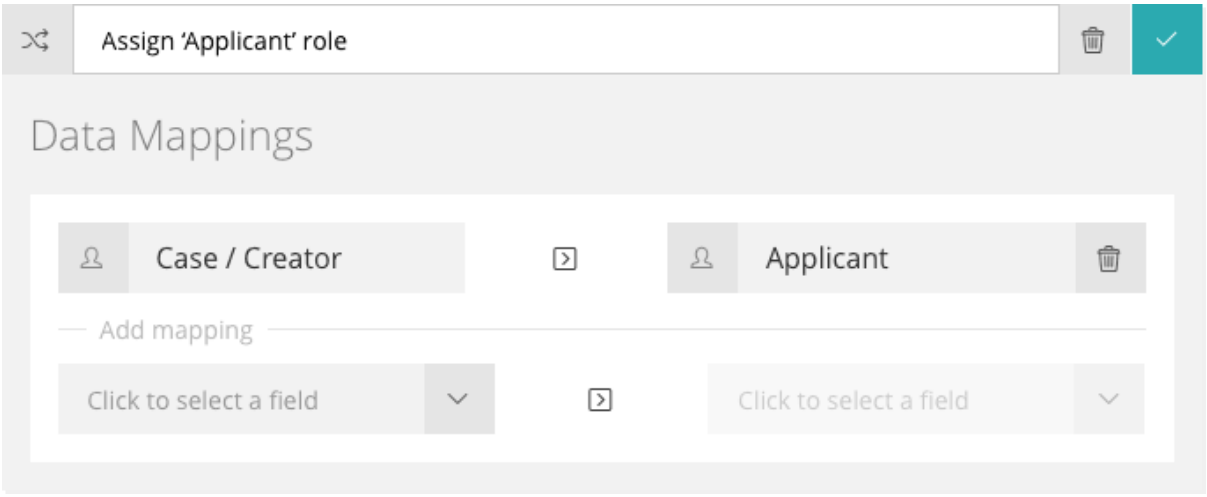
Signavio Process Manager defines these *Model state* options. See Managing approval workflows[17] for instructions on how to set this up.

7.14 Map variables

A "Map variables" task in the process editor

The "Map variables" action copies the value of one *variable* (page 43) to another. You can use this to set the value of a process variable automatically, instead of manually using a *form* (page 84).

In some processes, the person who started the case should participate in the process by adopting one of the process's *roles* (page 43). Consider an application process, where the person who starts the case must then complete a task to provide additional information.



Configuring a "Map variables" task in the process editor

In this example, the "Provide additional information" action assigns the task using an "Applicant" role. When the applicant starts a case, the "Map variables" task automatically sets this role to the value of the "Case creator". This automatically assigns the "Provide additional information" task, and any other tasks that use the same role, to the person who started the case.

7.15 Document template

New in version 3.32.

[17] https://docs.signavio.com/userguide/editor/en/workspace_admin/manage_software_config/config_workflows.html

If you are modeling a workflow, there is often the need to collect the information input entered by users of the workflow tasks. Workflow Accelerator can add these data dynamically to a Microsoft Word document, which is then used as a basis for a later audit, for example for further decisions. Because data are collected in a Word file, the content can be modified subsequently. To do this, add the action type 'Document template' in your workflow. With this action, a task is assigned to users including the request to specify the necessary information in a custom task form. Workflow Accelerator applies the information that is retrieved from this task form to the uploaded template.

Suppose your company has set up a workflow for contracts or quotations, in which several roles define the relevant document data in different Workflow Accelerator tasks. The contract or quotation is created based on a standard template, so that required data from the workflow are assigned accordingly in an output form. At the end of the process, the output document containing all relevant information is made available to a sales person, who has then the ability to make any personal additions before the contract or quotation will be sent to the customer.

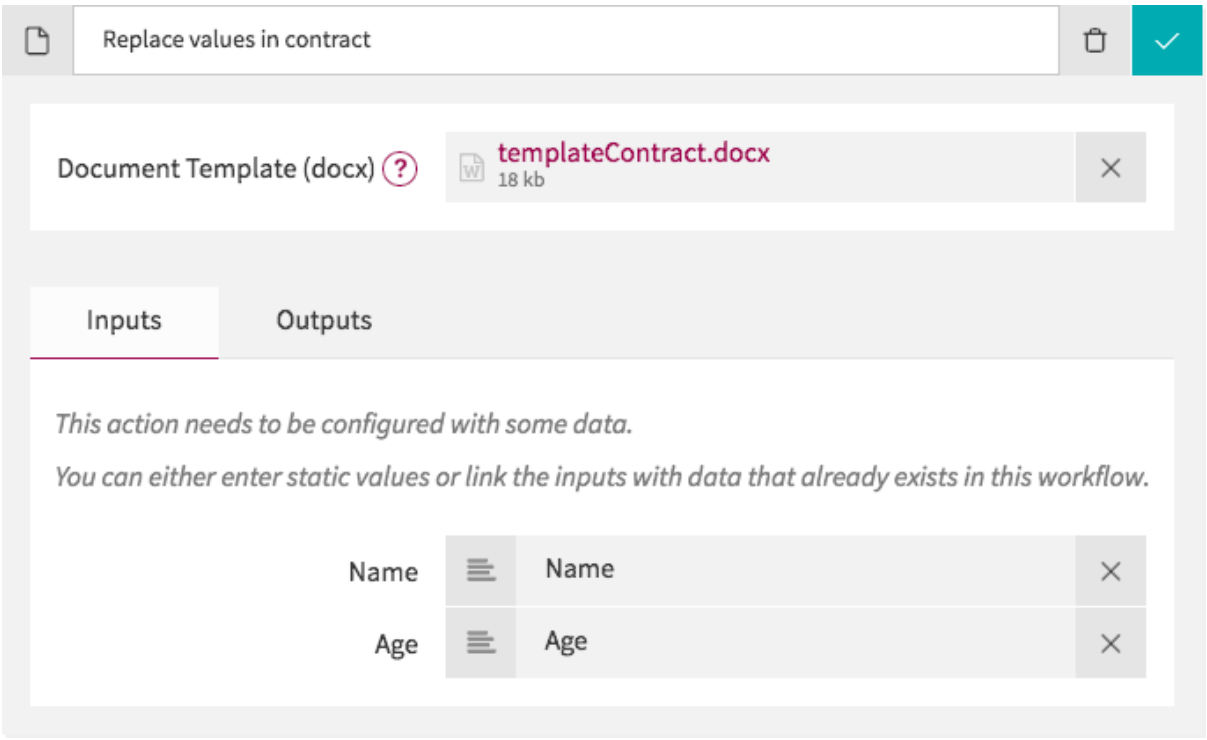
7.15.1 Content controls

A prerequisite for the implementation of this feature is that the applied template is a Microsoft Word document that contains the required content controls. You will find a detailed description on how to implement content controls in a Word document here: https://support.office.com/en-us/article/Create-forms-that-users-complete-or-print-in-Word-040c5cc1-e309-445b-94ac-542f732c8c8b

Hint: Please note that currently only text and plain text are supported as types of content controls.

Content controls, which have been created in the document template, each represent a form field. To map and display form fields correctly in Workflow Accelerator, it is mandatory to specify a title to each content control. The title is set as a property of the control.

Hint: Rich text fields don't support line breaks. If you need fields that support line breaks, use plain text fields and activate the check-box 'Allow carriage returns (multiple paragraphs)'.



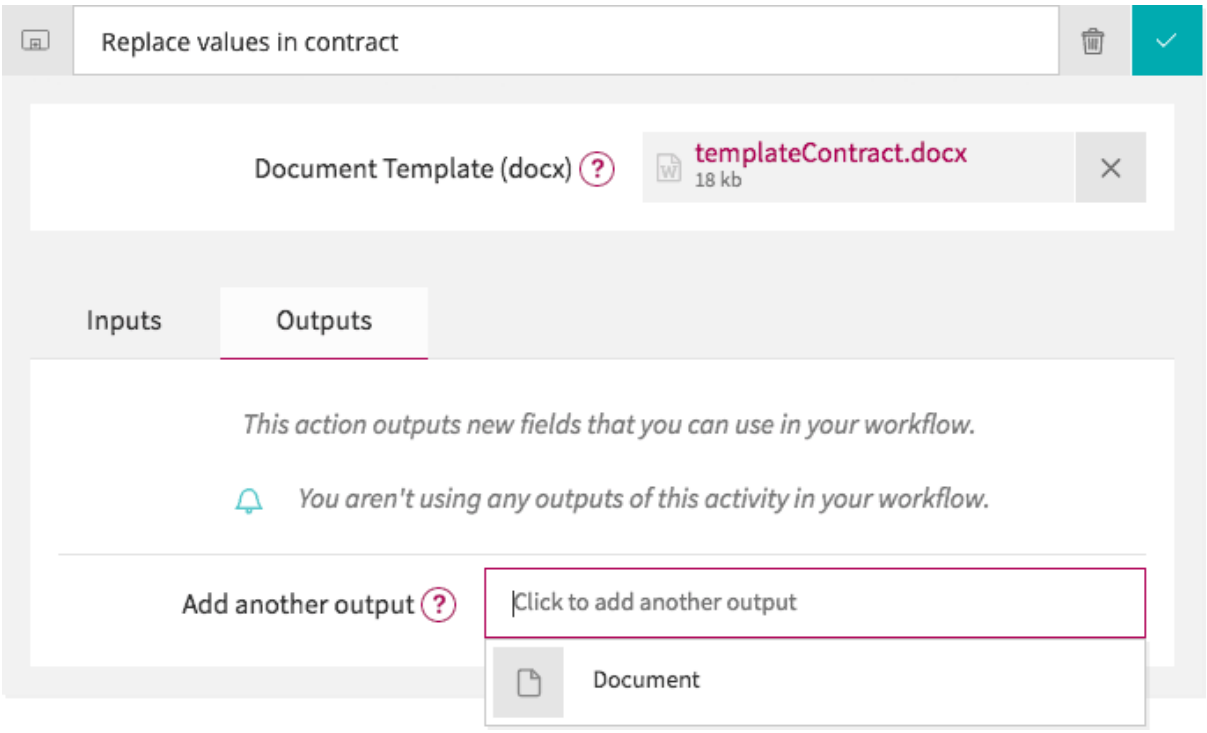
Configuring a Document template *action in the process editor.*

Select a *Document template* action in the process editor to configure the following options.

Document template Upload the template. This action creates a new file from a document template.

Inputs tab Specify the desired data and link either with static values or existing fields of the workflow.

Outputs tab Select the output value "Document" and enter the desired name.



Each field in the subprocess can be used in this process once you added it as an output.

Chapter 8

Forms

In Signavio Workflow Accelerator, you can use forms to enter information when you run a process. You can use forms in two places: form triggers and user tasks.

8.1 Form triggers

You use a *form trigger* (page 31) to set the values of *workflow variables* (page 43) when you start a new case for a process.

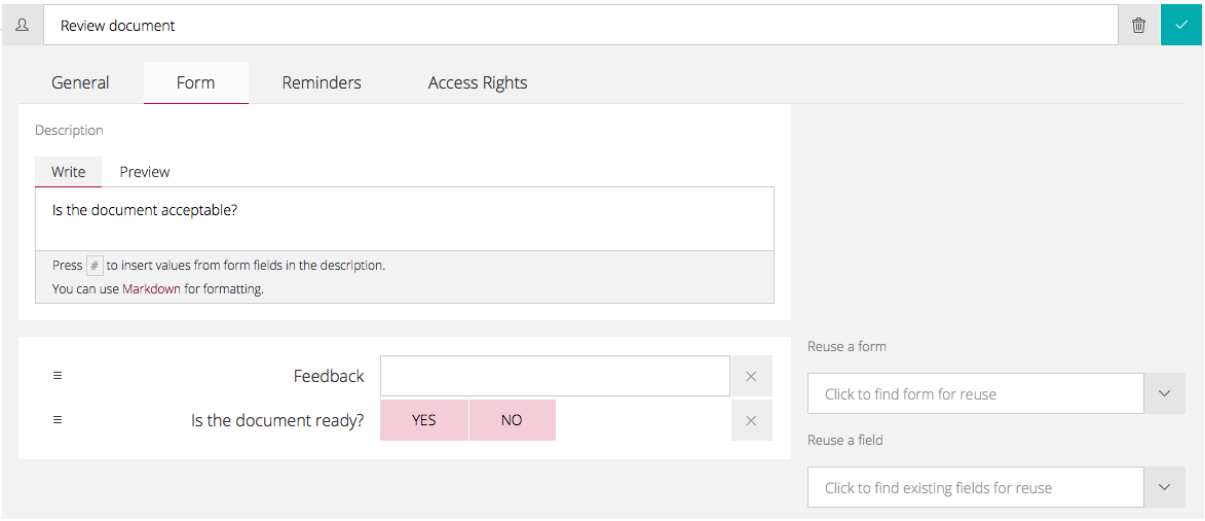
Starting a new case with a form trigger

This form has a description (“Enter personal details”) and two fields. The icon next to the “Name” field label indicates that the field has an additional description.

To add a form trigger to a process, use the process builder’s *Triggers* (page 30) page to select “When a form is submitted”.

8.2 User task forms

You use a *user task* (page 56) form to view and edit the values of *workflow variables* (page 43).

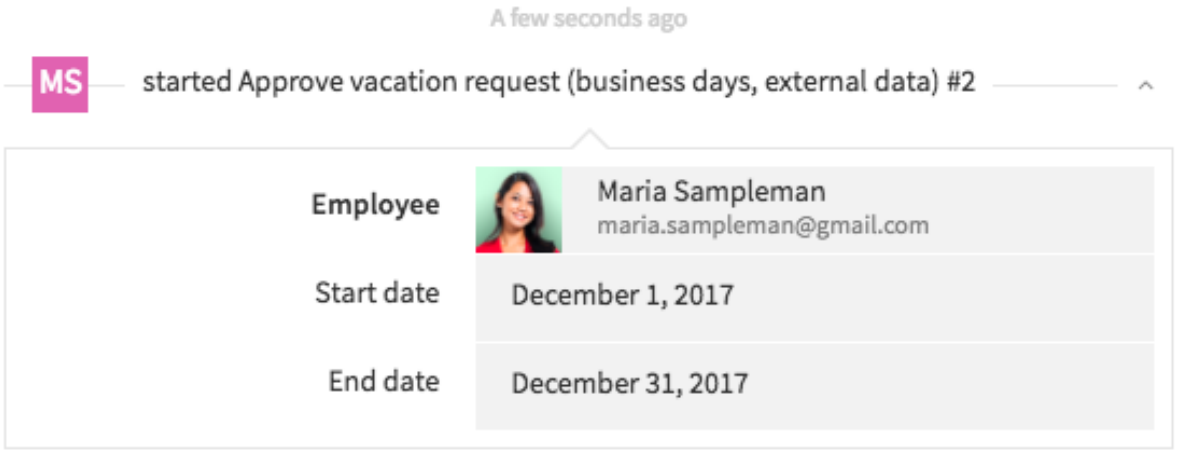


Viewing and editing values on a user task form

To add a user task form to a process, use the process builder's *Actions* (page 35) page to add a user task, and on the user task's configuration panel, select the *Form* tab.

8.3 Viewing form data

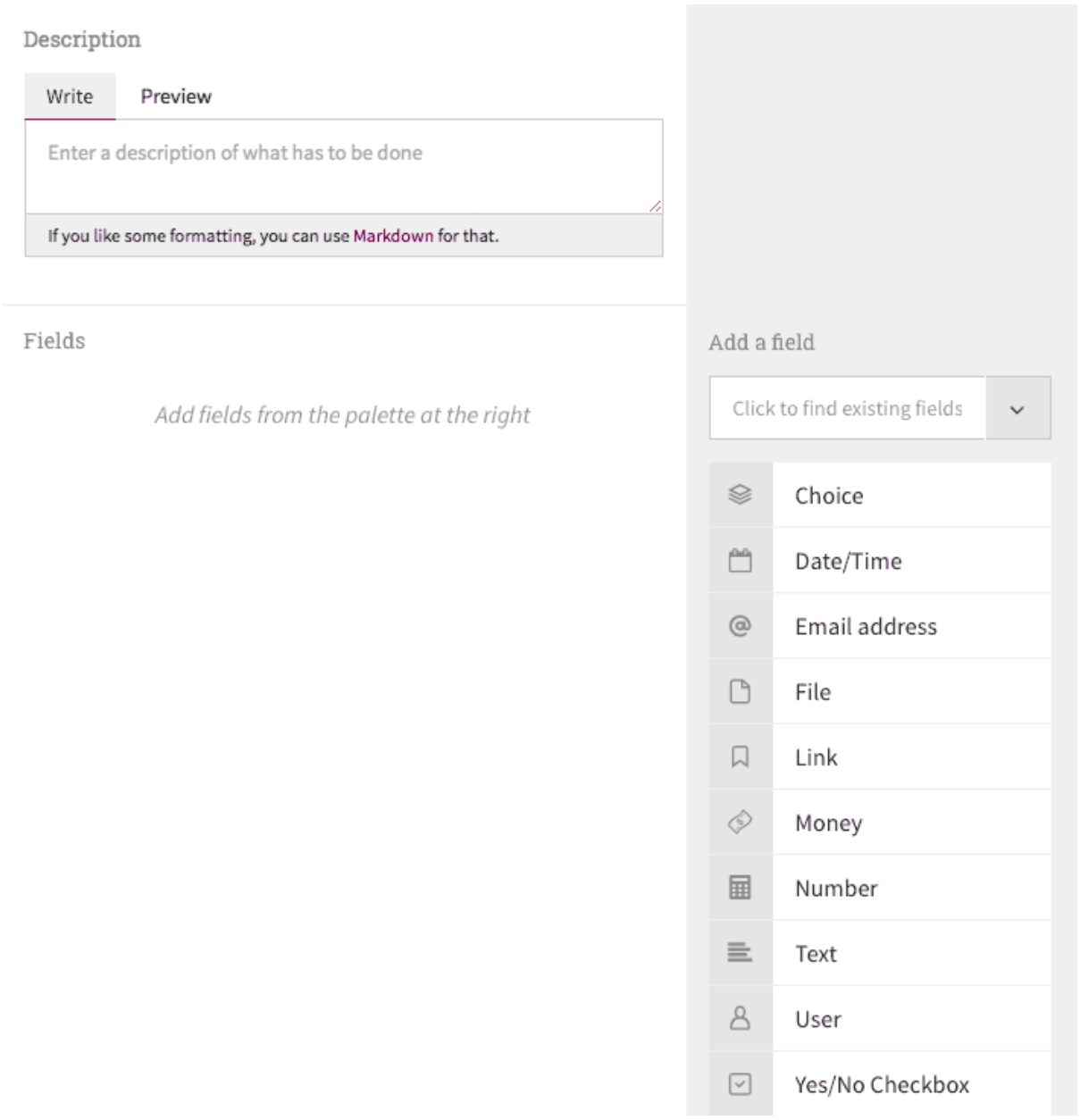
Form data can be found in the history panel. Click on the *History* button, then the **Form** tab to view all forms in an event. Click on the name of a form to view the form data.



8.4 Using the form builder

Use the form builder to create a from trigger or user task.

To create a form in the process editor, for a form trigger or user task, you use the form builder.

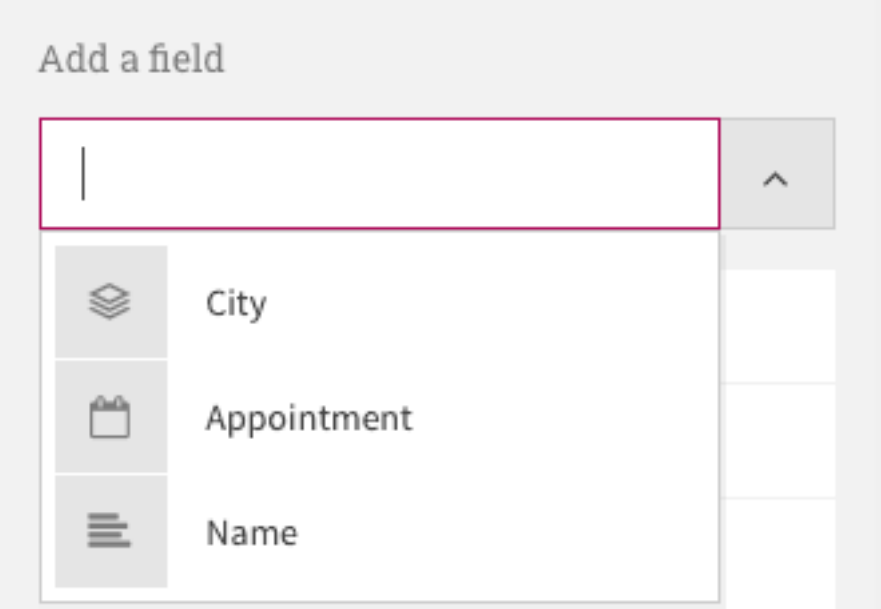


The process editor's form builder for a new empty form

Use the "Description" text area to provide initial instructions to people who complete the form. You can use *Markdown* (page 129) to format the description, which makes it convenient to link to additional information, for example.

The "Add a field" section contains a field types palette. Click one of the field types to add a field of that type to the form.

The "Reuse a field" section lists variables that the process has already defined on other forms, such as a trigger form:



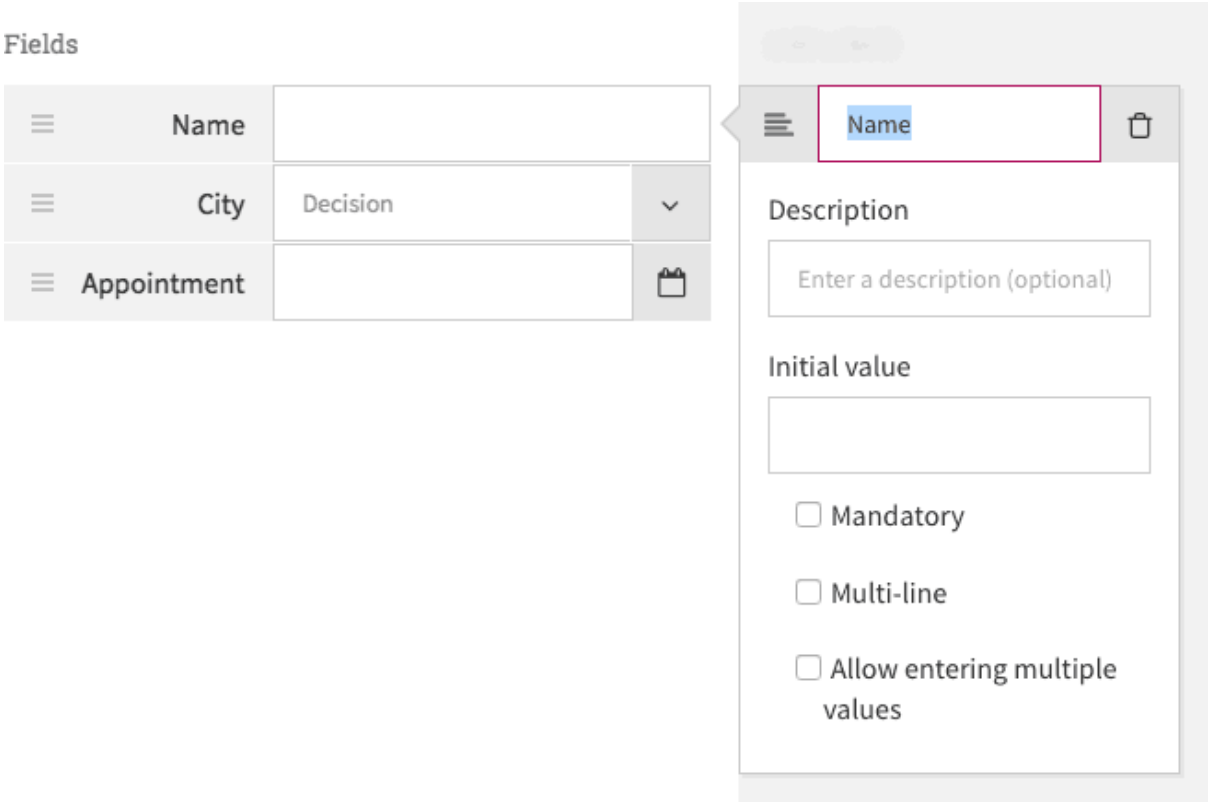
Adding an existing workflow variable to a form

Adding an existing field to a form makes it possible to view or update existing information, such as a user task form that you use to complete information the person who started the case did not enter on the trigger form.

You can also choose to reuse an existing form from a previously created user task or form trigger. Doing so copies all fields and field configurations (including mandatory, read-only and custom conditions). If you later edit the original form field, it will not change the information in the duplicated form.

8.5 Configuring form fields

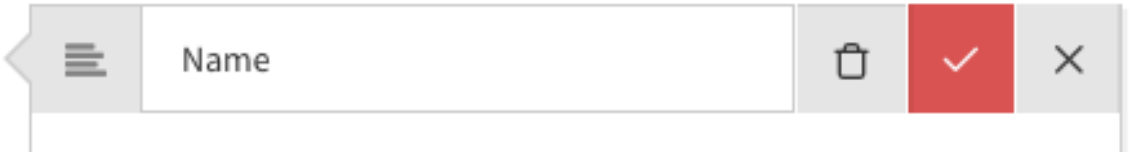
The “Fields” section shows the fields you add to a form:



Configuring a form field in the process editor's form builder

Select a field in the "Fields" section to open its configuration panel. The top row of the field configuration panel shows the field type label, the editable field name.

To delete a field, click the delete button in the top-right corner, and click the confirmation tick mark button to confirm:



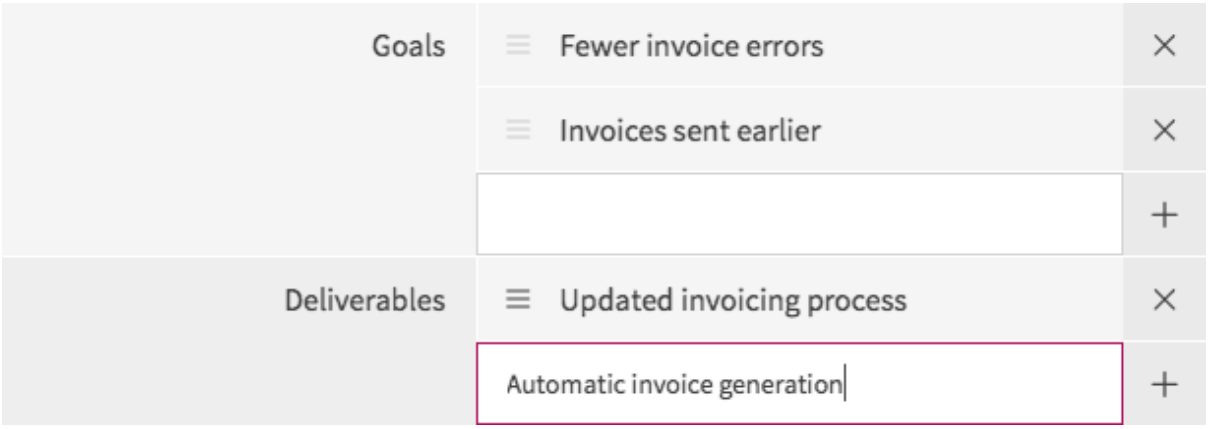
Deleting a form field

You can also configure the following properties.

- Description - an optional field description, shown via context-sensitive help icon
- Initial value - an optional default value that pre-populates the form field
- Read-only - specifies that you cannot edit the value, used to display previously-entered information
- Mandatory - specifies that you must enter a field value, so that you cannot complete the form without a value for this field
- Allow entering multiple values - specifies that the field has a list of values that you add and remove independently

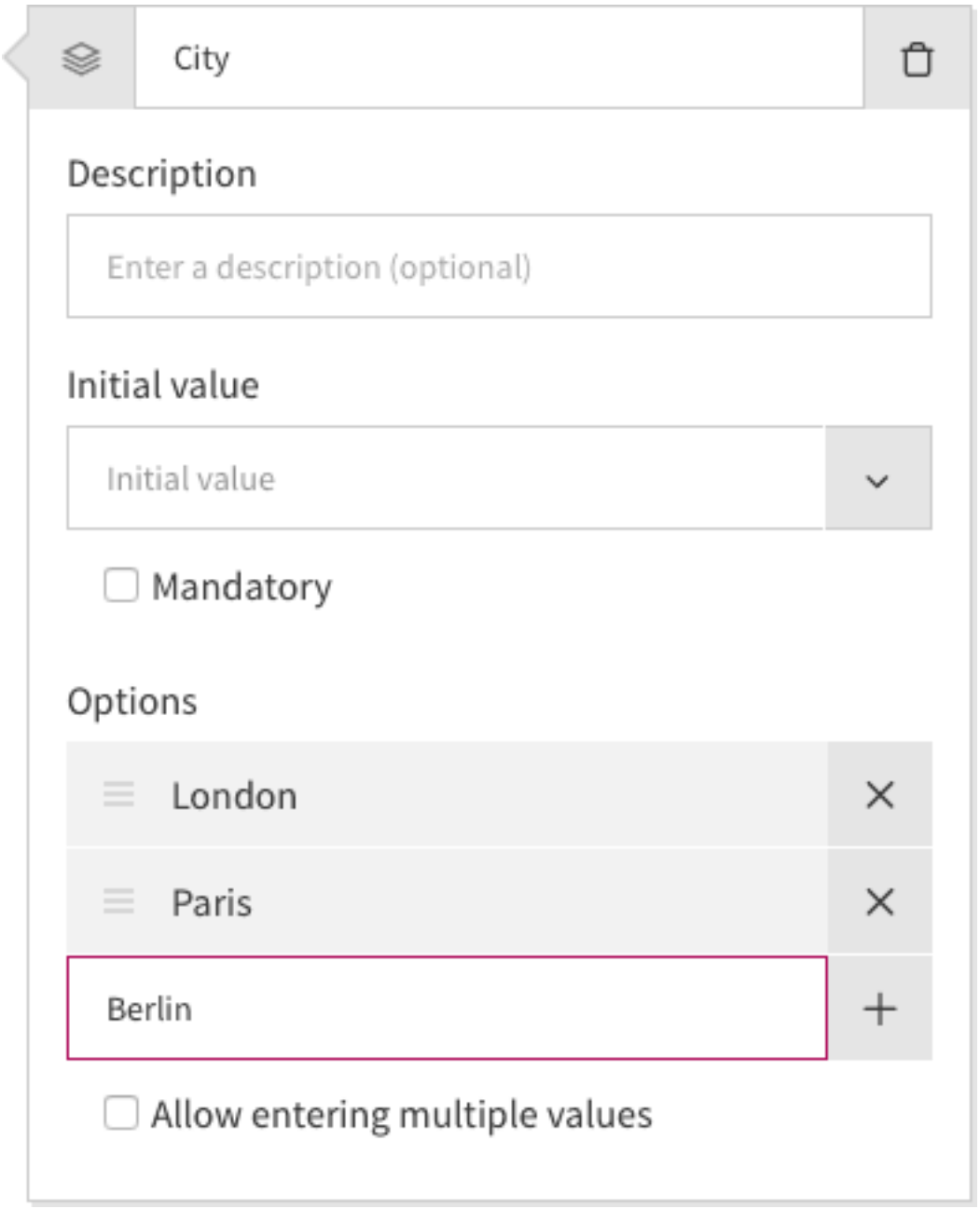
Text fields also have an additional "Multi-line" option that configures Workflow Accelerator to display the form field as a multi-line text input area, for longer text values.

When you enable the “Allow entering multiple values” option, the field appears differently on forms. Entering a value adds it to the list. Use the × icon to remove a value.



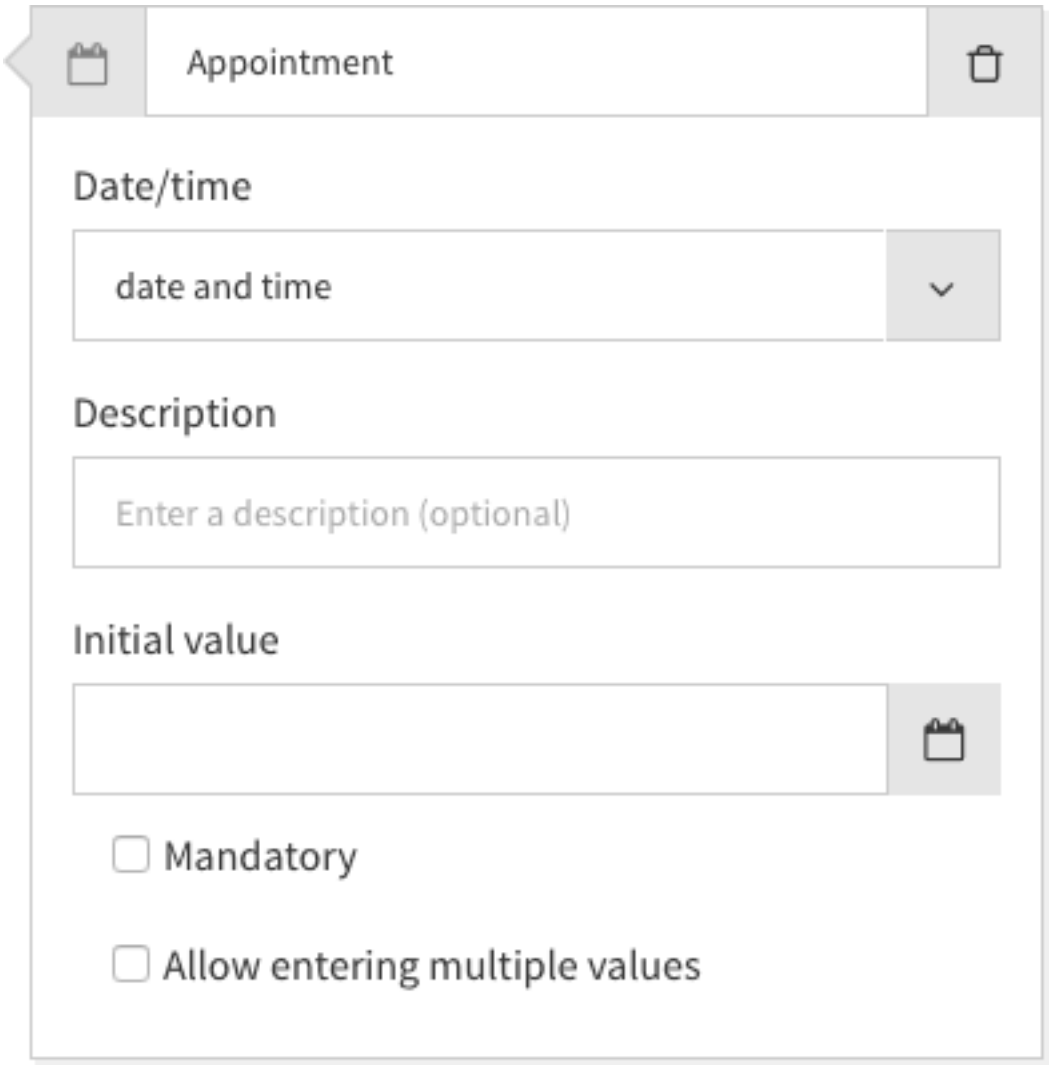
Entering multiple field values in two text fields

Choice fields have a “Options” - the list of values to choose between:



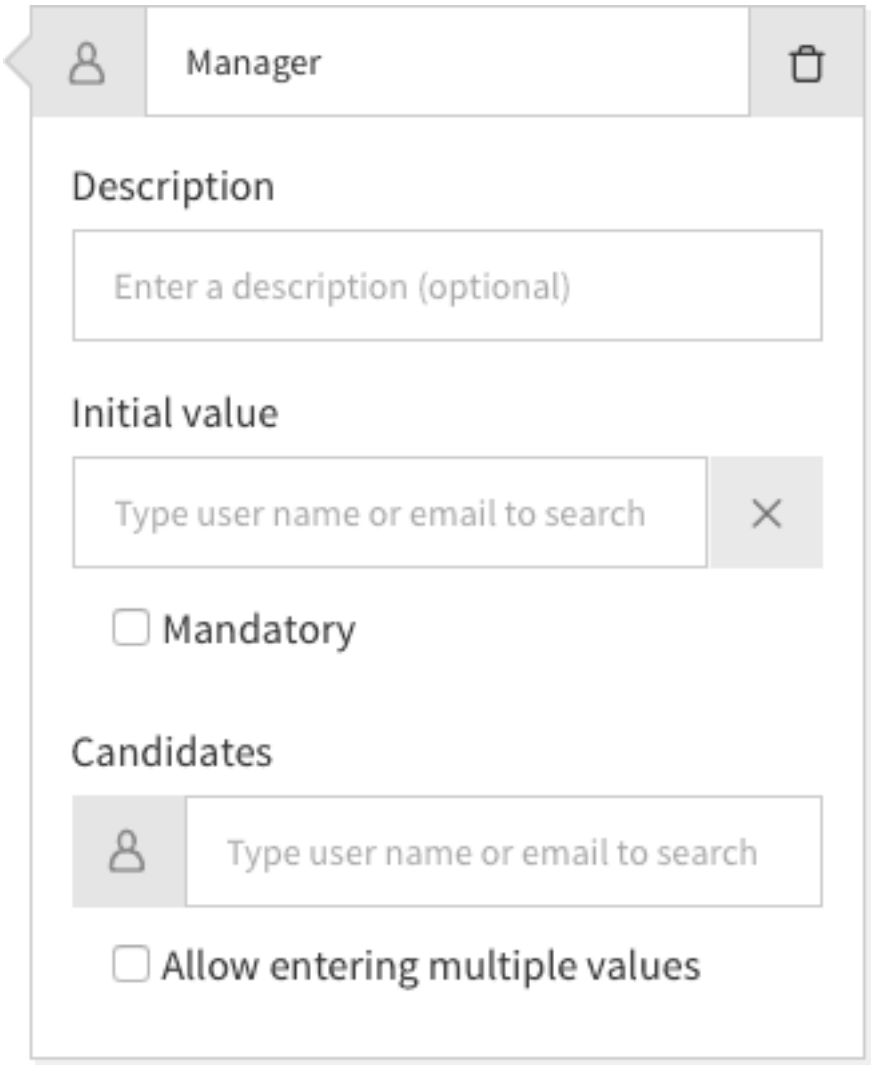
Choice field configuration

Date fields have a “Date/time” option for choosing between a date and time, just a date without a time of day, or just a time:



Date field configuration

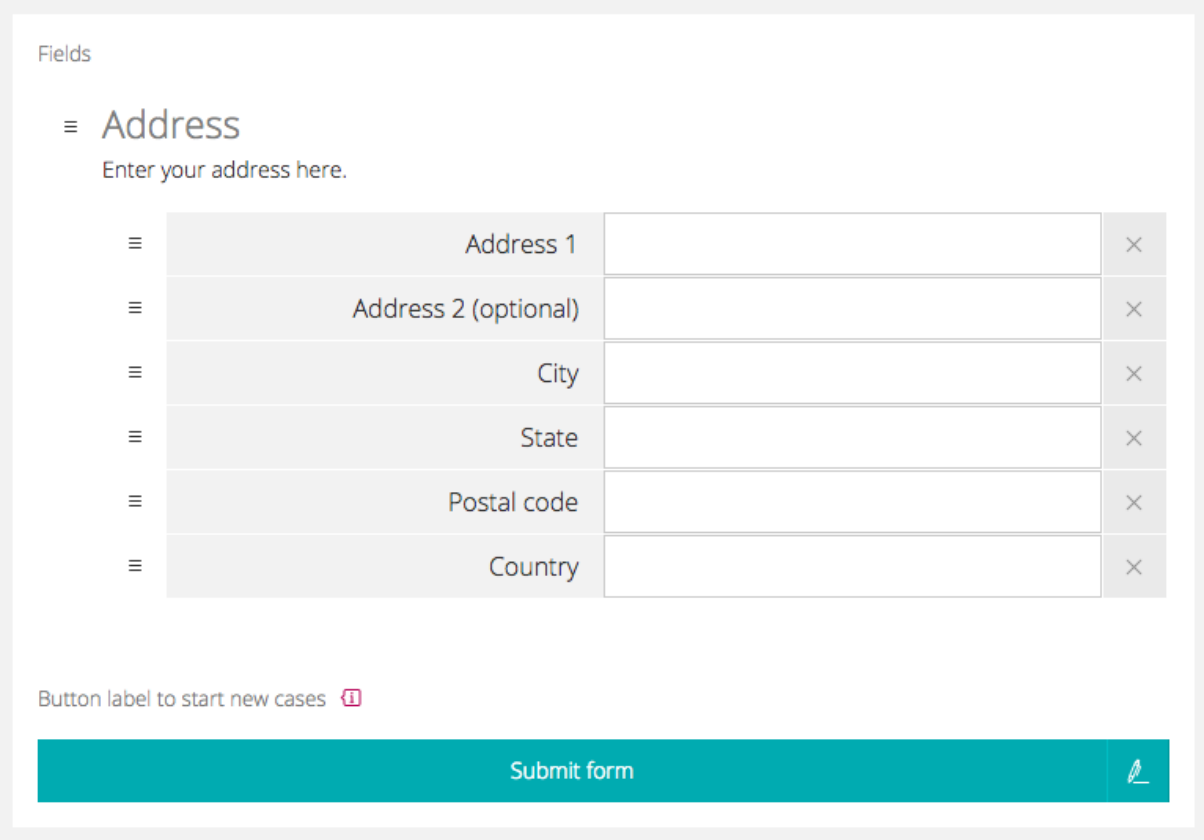
User fields also have a “Candidates” option that you can use to choose which users you can assign:



User field configuration

8.6 Form groups

If you have a very long form, keeping your fields as a long, unbroken list can be confusing. You can use form groups to add structure to your form by grouping similar fields together. For example, you can use form groups to create a section where users can add their address.

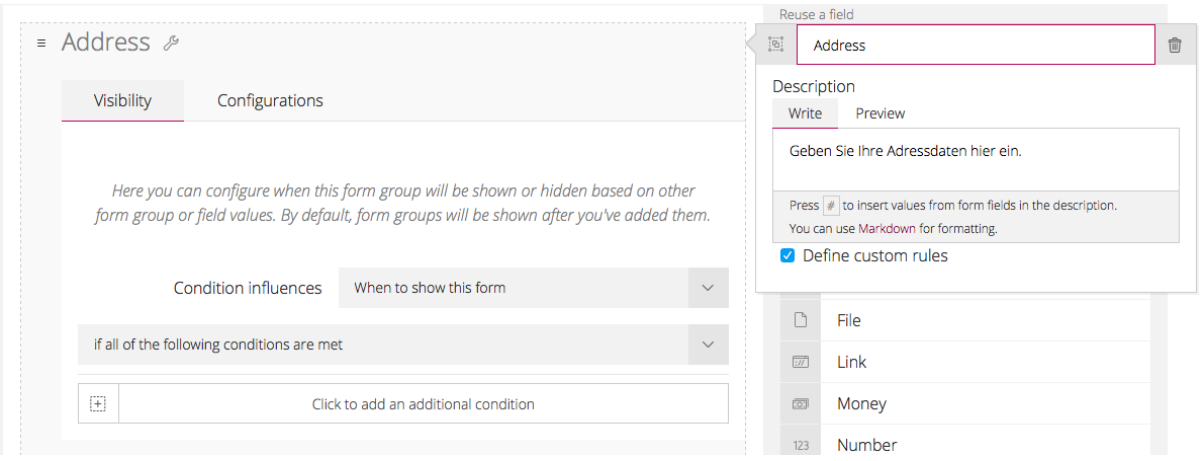


Form groups can be used to create sections, such as this one for address information

To use form groups, navigate to the process builder and scroll down to the form builder. Click the "Form Group" button underneath the field types palette on the right-hand side. In the configuration dialog, you can name your form group and write a description, if desired. (You can use *Markdown* (page 129) to add formatting your description.) From there, simply drag and drop fields from the list to add them to your form group.

Hint: Form groups can be reused in other forms. They can also be nested inside other form groups, like a folder.

You can define custom rules for your form group, such as when the form should be shown to users. These rules are applied to all fields within the form group, unless the field has a custom rule of its own.



Setting a custom rule for a form group

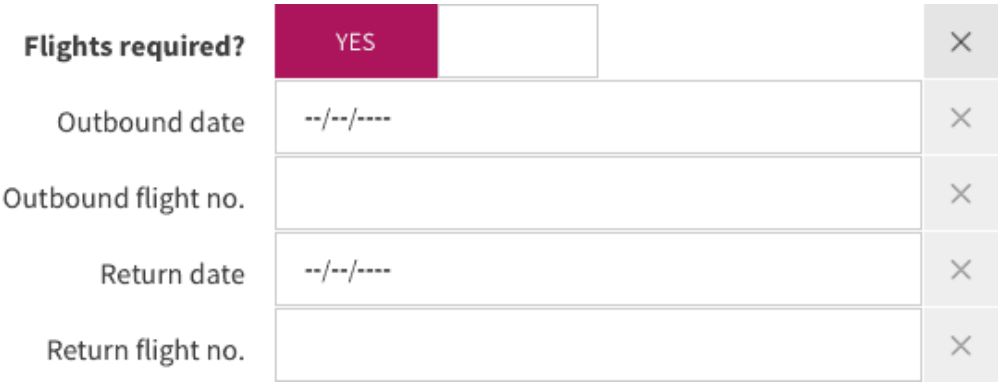
8.7 Dynamic form fields

New in version 3.40.

In the *Arrange Business Trip* process example, the *Book flights* user task records whether the trip requires flights.

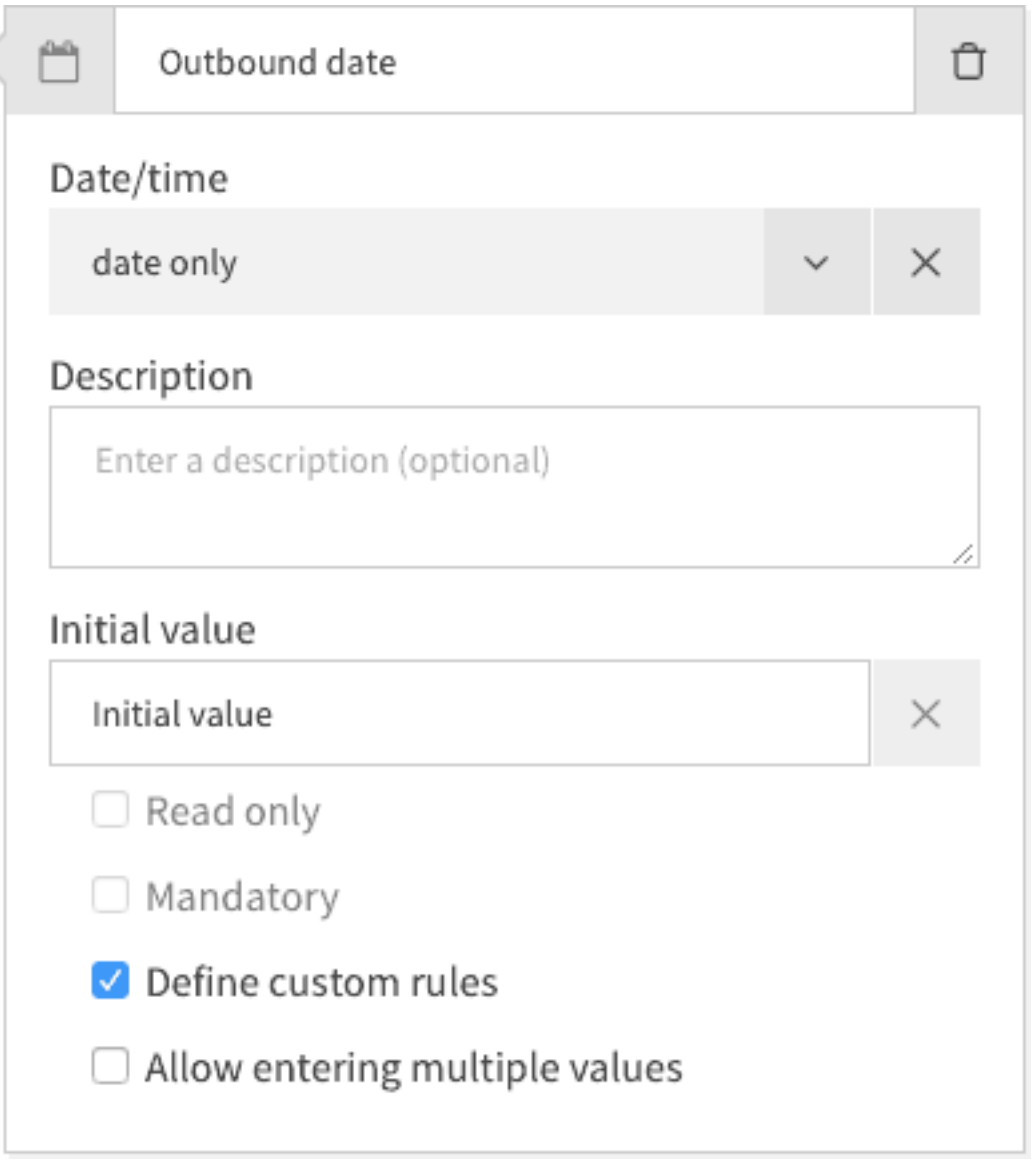
A selection that does not *require flight information fields*

When a trip requires flights, additional flight information fields become relevant.



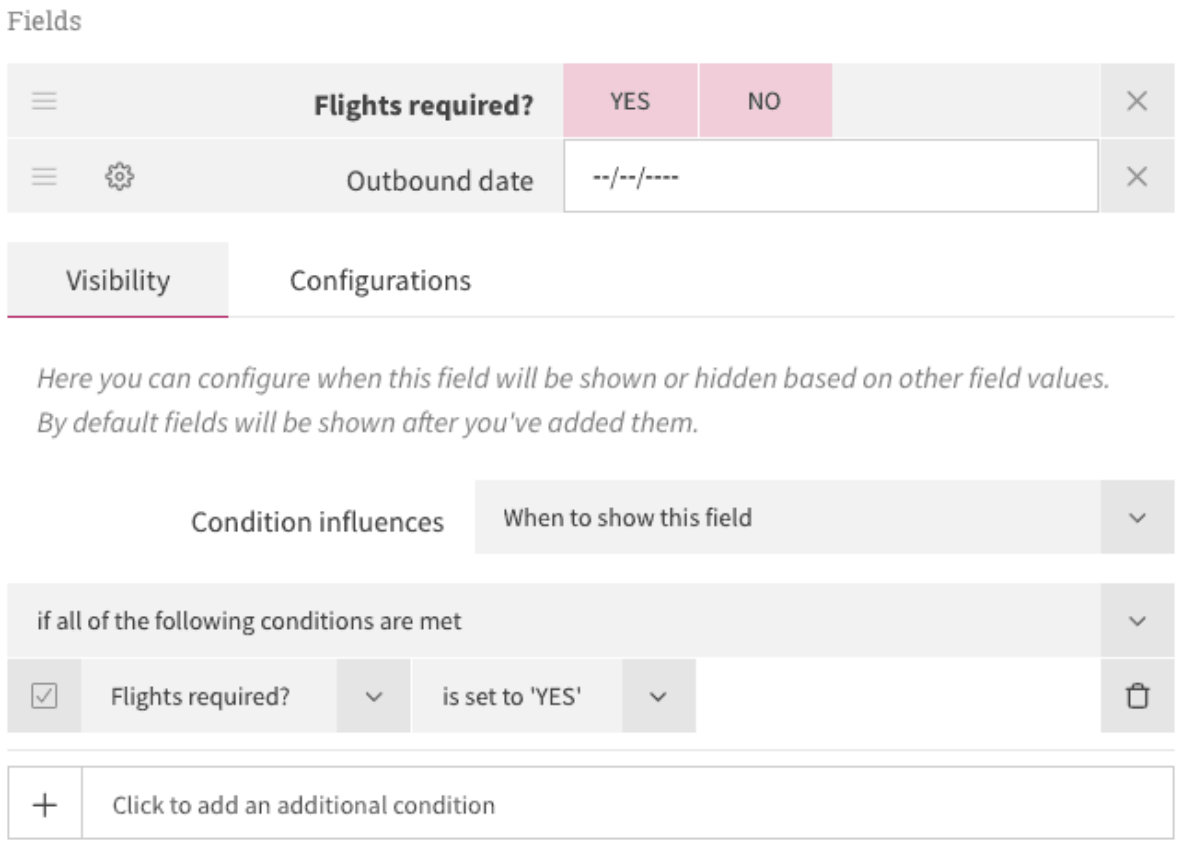
A selection that does *require flight information fields*

Sometimes, a process only uses a form field when another field has a certain value. You can configure form fields with a dynamic configuration that depends on custom rules.



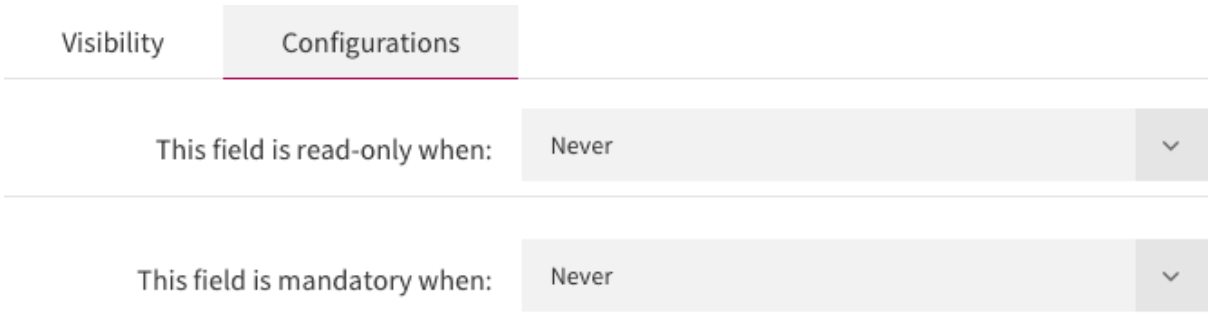
The Define custom rules *form field configuration option*

When you select *Define custom rules*, you can specify conditions that determe when the form shows the field.



Configuring custom visibility rules for the Outbound date *field.*

You combine multiple conditions and use them to determine whether to show or hide the field.



Configuring custom field options

On the *Configurations* tab, you can also configure conditions that make the field read-only or mandatory.

Chapter 9

Control flow

You use transitions, gateways and events to specify the processing order of the actions in a process.

9.1 Transition

The *process builder* (page 28) displays a transition an arrow from a source element to a destination element. The transition specifies that the workflow engine only executes the destination element after completing the source element. BPMN calls a transition a “sequence flow”.

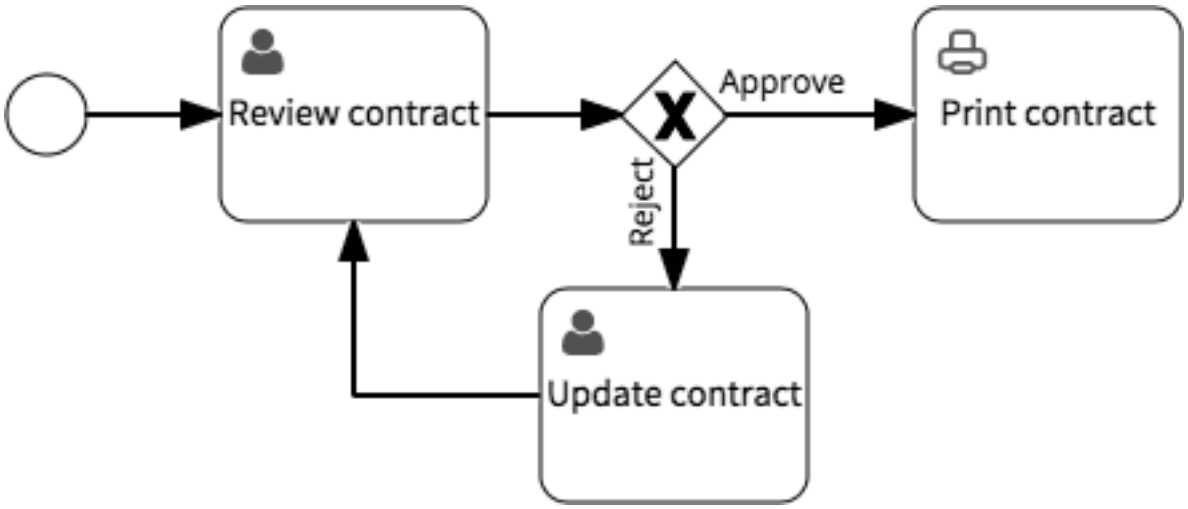
9.2 Exclusive gateway

Use an exclusive gateway to make a choice between multiple execution paths. The exclusive gateway selects one of the outgoing transitions, and only continues execution on that transition. You can configure an exclusive gateway with a *manual decision* or an *automatic decision*.

9.2.1 Manual decision

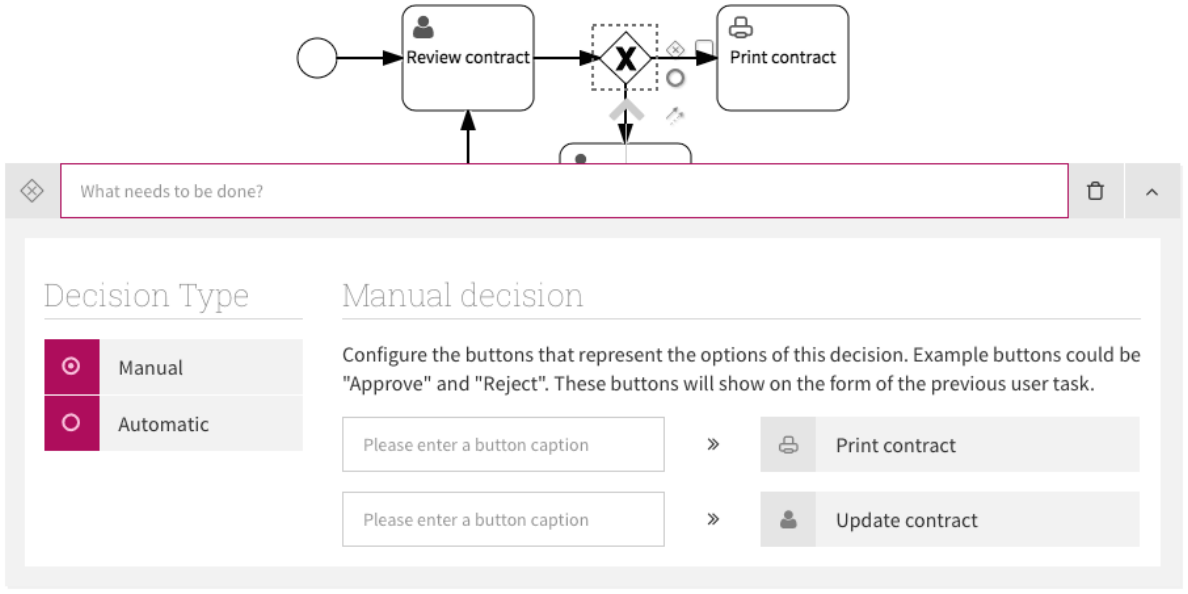
Use a manual decision for an exclusive gateway when a person must make a decision. A user task must precede the gateway; this task includes making the decision. The user interface presents the decision to the user as buttons on the user task form.

Suppose you have a user task called “Review contract”, an exclusive gateway and the two user tasks “Print contract” and “Update contract”:



An exclusive gateway must have at least one incoming and two outgoing transitions

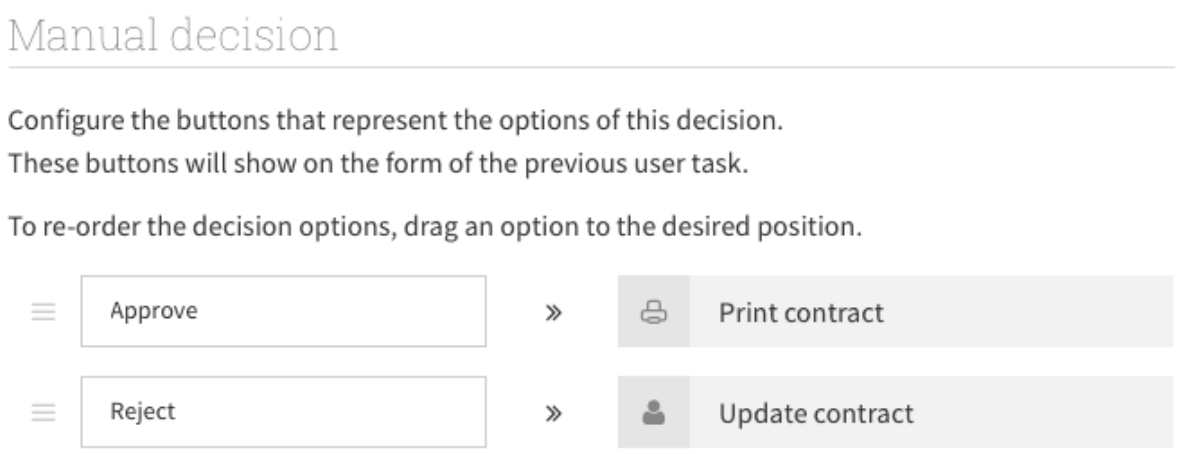
Select the exclusive gateway. Its type defaults to *manual decision*. After creating the elements and connecting them, as above, you have configured the exclusive gateway:



Default manual decision configuration

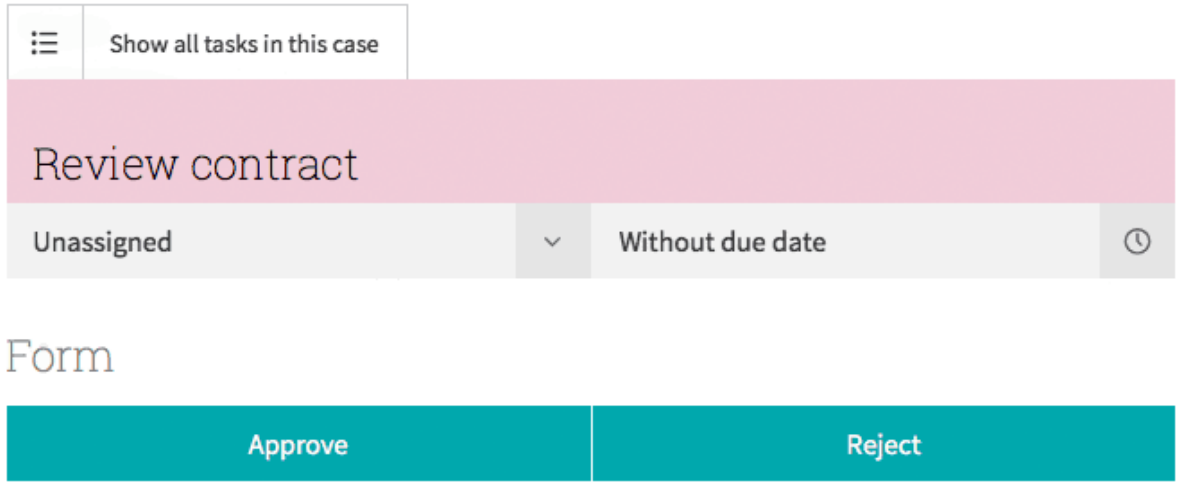
In order to use the decision you need to name the buttons which will represent the decision. For each button, the label on the right indicates the next action in the process, which Signavio Workflow Accelerator will perform when someone clicks the button. In this example, when the user clicks the decision button “Print contract”, Workflow Accelerator executes the “Print contract” task, but *not* the “Update contract” task.

You can easily change the text on the buttons, and order they appear in. For example, change them to “Approve” and “Reject”, and drag the “Approve” button configuration to the top so that “Approve” appears first on the form:



Customized decision buttons

After starting a new case for this process, the “Review contract” task will have decision buttons:



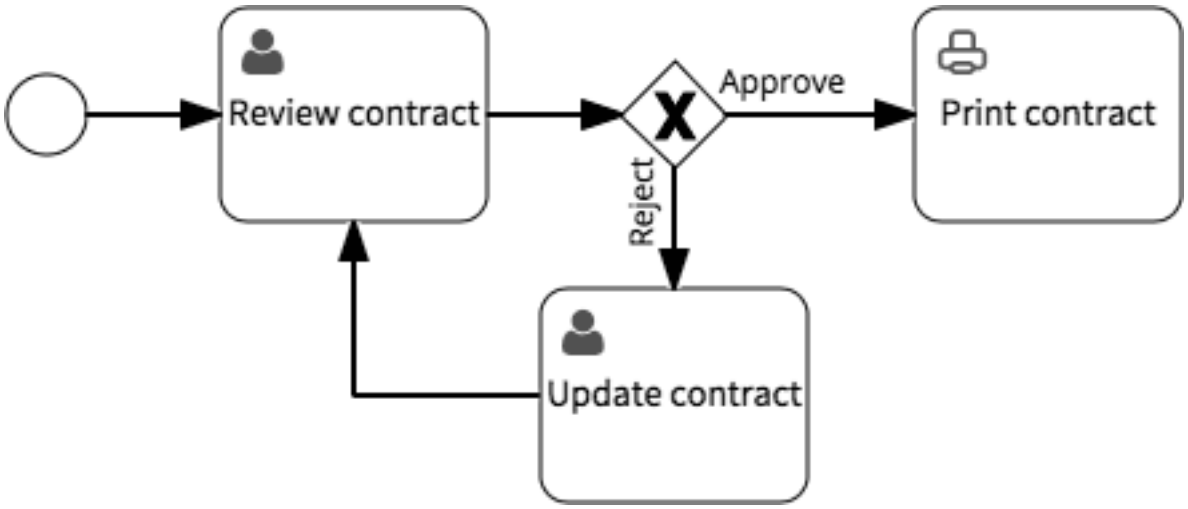
Task decision buttons

When the task before the exclusive gateway - “Review contract” in this example - has a form, the form includes the decision buttons.

9.2.2 Manual decision variable

Adding a manual decision to a process also creates a *workflow variable* (page 43). You can use this to re-use the result of a decision later in the process, either to display the entered value on another form, or to use the value in an automatic decision’s condition.

During workflow execution, selecting a decision sets the variable’s value to the selected decision - the text on the decision button. In this example, the decision variable has the value “Approve” or “Reject”.

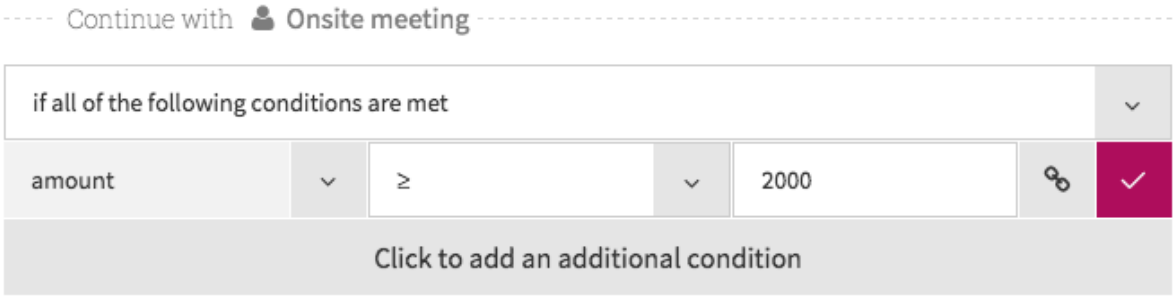


Decision variable values - "Approve" or "Reject"

The variable has the name "Decision", by default, or the name of the gateway if it has one. You can change the variable name on the process editor's "Details" tab, in the "Field overview".

9.2.3 Automatic decision

An exclusive gateway that selects an outgoing transition based on conditions that you choose models an automatic decision. For each transition, you can formulate a condition using workflow data. The workflow engine evaluates transition conditions in order, from top to bottom. The workflow engine will take the transition with the first condition that evaluates to true, using the current case's field values.



Automatic decision condition editor

To specify a condition, start by selecting a field and a comparison operator. Enter either a static value in the input field on the right, or click the button to select another field.

A condition can include multiple field value comparisons. To add more sub-conditions, click the button at the bottom of the list. You can also use the select field at the top to specify that either **all** conditions in the list must evaluate to true, or that at least one of them must evaluate to true.

If you do not completely specify a sub-condition, evaluating the whole condition will fail and the workflow engine will not follow the transition. The symbol indicates an incomplete sub-condition, while the symbol indicates a valid sub-condition. Click either of these symbols to remove the sub-condition from the list.

Default transition

An automatic decision usually has a default transition. You use a default transition as a fallback mechanism: if none of the conditions evaluate to true, the workflow engine follows the default transition.

The default transition

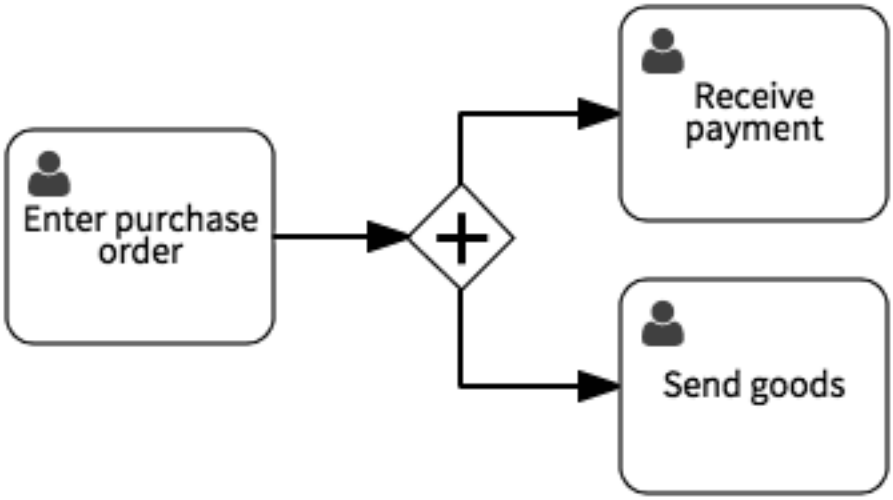
To make a transition the default, select the "per default" item in the selection field at the top.

9.3 Parallel Gateway

Use parallel gateways to model tasks that people will complete at the same time as each other, or one at a time but not in a particular order. To do this, you *fork* and *join* the sequence flow.

9.3.1 Forking

With a parallel gateway, you can fork execution into multiple, concurrent flows. When process execution arrives in a parallel gateway, the workflow engine creates a new individual execution flow for each of the gateway's outgoing transitions. Let's look at the following purchase order example:



A parallel gateway example

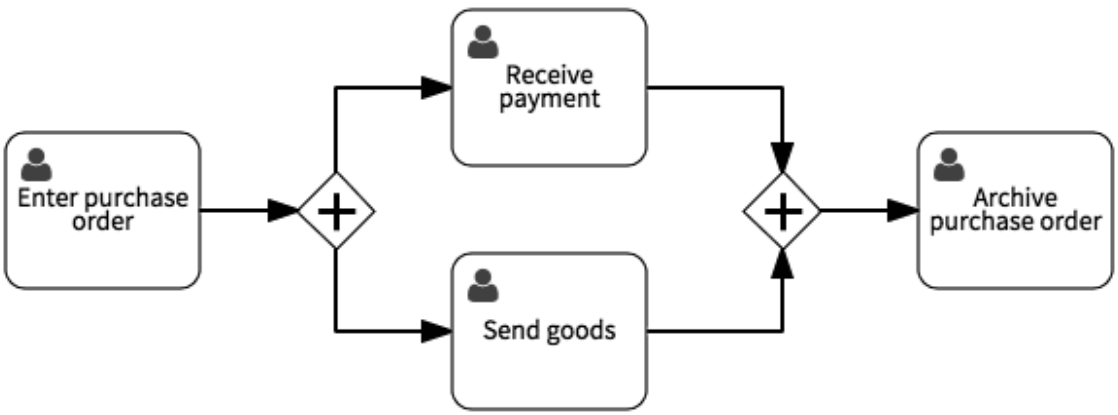
In this example, completing the “Enter purchase order” user task activates the parallel gateway. The parallel gateway will create two individual paths of execution. One will take the transition to “Receive payment” and create that user task. Meanwhile, the other will create the “Send goods” user task.

You can have as many outgoing transitions as you want. The workflow engine will create all destination tasks for those transitions at once.

9.3.2 Joining

You also use a parallel gateway to join concurrent paths back together. In this case, the joining parallel gateway has more then one incoming transition. Workflow execution will wait at the gateway until as many execution flows arrive as it has incoming transitions. When the last concurrent flow arrives, the joining parallel gateway will activate and the workflow engine will create one execution flow on the outgoing transition.

To continue the previous example, extend the purchase order process to look:

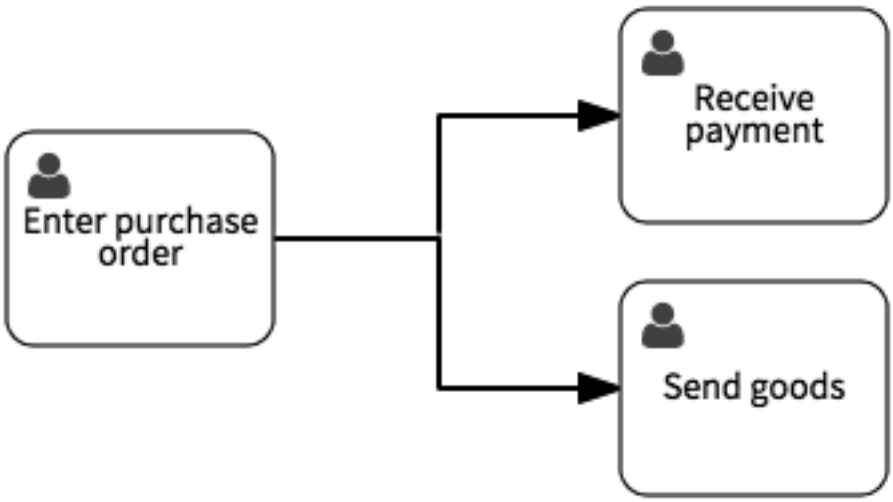


A parallel gateway example with join

In this example, “Archive purchase order” will only start after people complete both the “Receive payment” and “Send goods” tasks.

9.3.3 Default forking

By default, the workflow engine interprets multiple outgoing transitions from an action as parallel tasks. This means that if you have multiple transitions from a user task, the workflow engine will create concurrent tasks for all of the transitions’ destination actions. Let’s look at a simple example.



Default forking

After “Enter purchase order” completes, the workflow engine will create the tasks “Receive payment” and “Send goods” immediately.

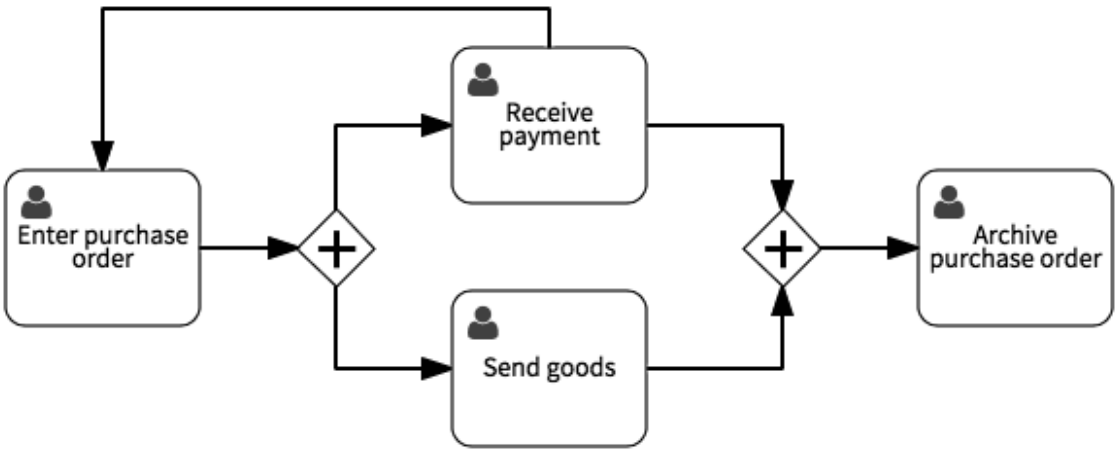
You can combine default forking with a parallel gateway for joining.

9.3.4 Default merging

When multiple transitions lead to a user task, the workflow engine will start the user task once for each execution flow that arrives. This means that the workflow engine does not perform implicit joining for parallel flows.

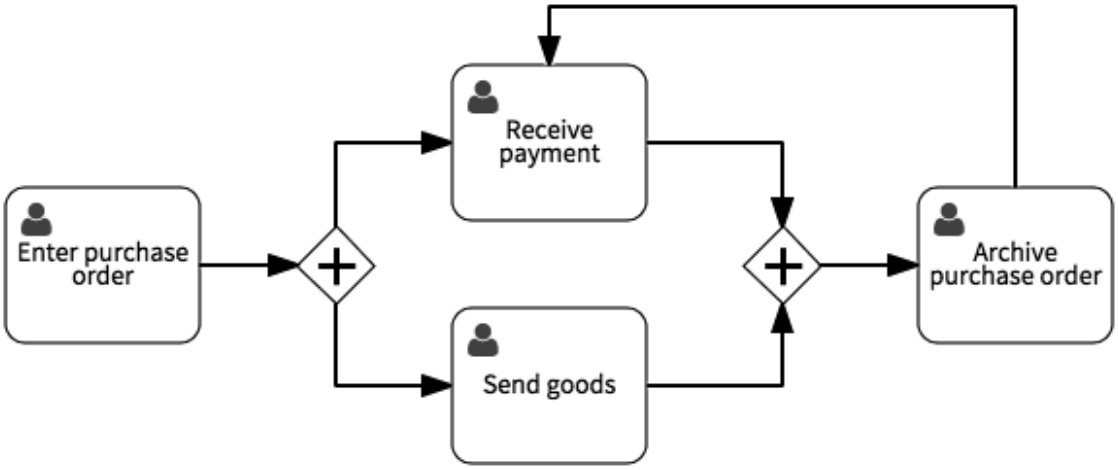
9.3.5 Parallel gateway issues

You will end up with problems if you loop back over parallel gateways. To avoid situations:



Undesirable loopback

and this:



Undesirable loopback

To avoid these issues, think of all actions between forking and joining as a self-contained part of the process, such that no transitions should cross that scope.

9.4 Start event

A start event marks the start of a process. All process elements that do not have incoming transitions act as start elements. Start events don't have a direct connection to triggers. You can usually leave out start events if you want to create more concise diagrams.

9.5 End event

Like start events, you can also omit end events. End events mark the end of an execution flow:

9.6 Intermediate timer event

An intermediate timer event indicates that process execution waits for a timer. You can use this to prevent Workflow Accelerator creating the next task in a process until it becomes relevant.

Using an intermediate timer event to model an evaluation period

Configure how long the timer waits by selecting the timer in the process editor. In an open case, you can *skip a timer* (page 25) manually.

9.7 Intermediate link event

The intermediate link event allows you to trigger other processes. It is similar to the sub-process activity, except it does not wait for the sub-process to execute before continuing the parent process.

9.8 Milestone

A milestone is an intermediate event which allows you to mark an important event or a turning point within a process. By setting milestones, process owners obtain an overview of the workflow progress.

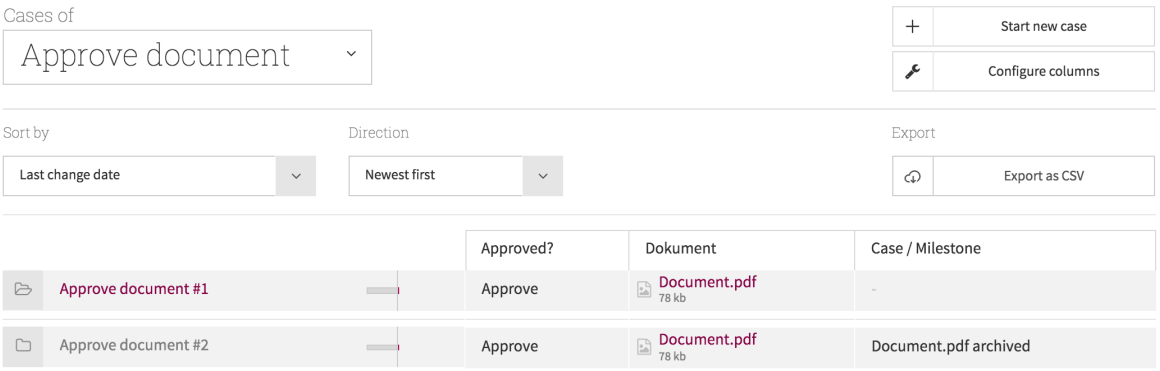
You can set a milestone either by using the intermediate event or via a script task.

Script task sample:

```
_case.milestone = 'Document archived'
```

When using the intermediate event, you can reuse any variables from the workflow to create the milestone text by typing #.

To show the current milestone, add the field “Case/Milestone” as a column in the case list.



Milestone overview

Note: Please keep in mind that only the latest milestone is displayed.

Chapter 10

Access control

You can use access control in Signavio Workflow Accelerator to restrict who can access a process as a whole, or specific tasks within a process. Processes and tasks default to public accessibility, which means that all users in the organization have access. When you configure access controls, you restrict access to specific users or groups.

10.1 Restricting access to processes

When you make a process private, you can grant six different permissions to users and groups.

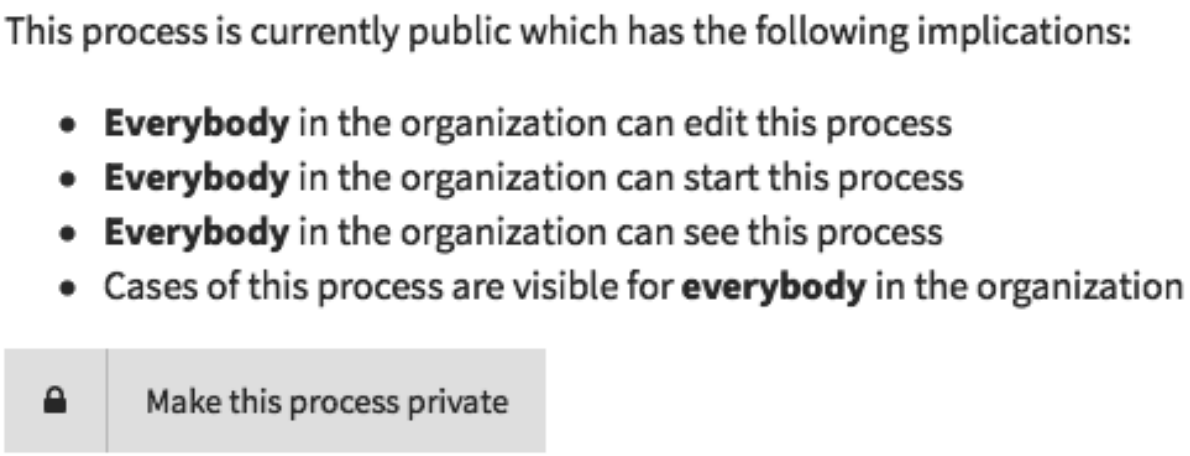
1. *Edit process* - make changes to a process and publish new versions
2. *Start process* - start new cases for the process
3. *View process* - see the process in the list of processes
4. *Edit cases* - work on the process' cases, by editing or completing tasks
5. *View cases* - see the cases for the process
6. *Create reports* - create reports of the process.

You can use these access controls in several ways, to restrict how people work on processes. For example, use the permission:

- *Edit process*, granted to a group, to restrict process editing to experienced process modelers
- *View process*, granted only to your own user, to hide incomplete or draft processes from other people while you create a first version
- *Start process*, granted only to your own user, so that people with *View process* and *Edit process* permission can collaborate on process modeling but cannot start cases until you publish it
- *Edit cases*, assigned to one group but not another, to allow one group to work on cases, while the other group can view their work.
- *View cases*, assigned to a group, to restrict access to cases that contain sensitive information,
- *Create reports*, granted to a business user group to allow them to analyze process metrics.

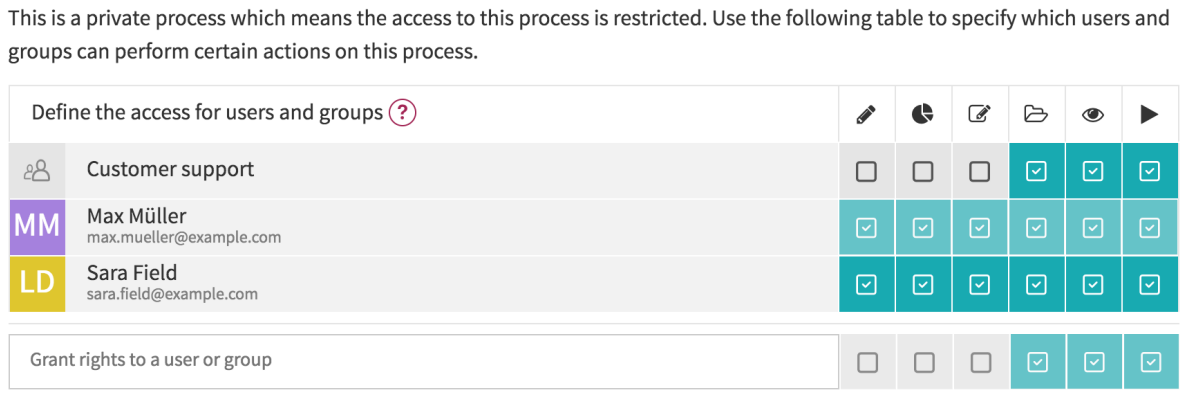
To apply process restrictions, open a process and select *Details*.

On the *Process details* page, the *Options* tab has an *Access rights* section.



Process details - access rights

Click the *Make this process private* button to configure access control.



Configuring process access control

You can now use the checkboxes to grant permissions to users and groups. Use the text box to search for additional users and groups, to add them to the table, so you can then grant access to them.

Click the *Make this process public* to remove all access restrictions on the process.

10.2 Restricting access to user tasks

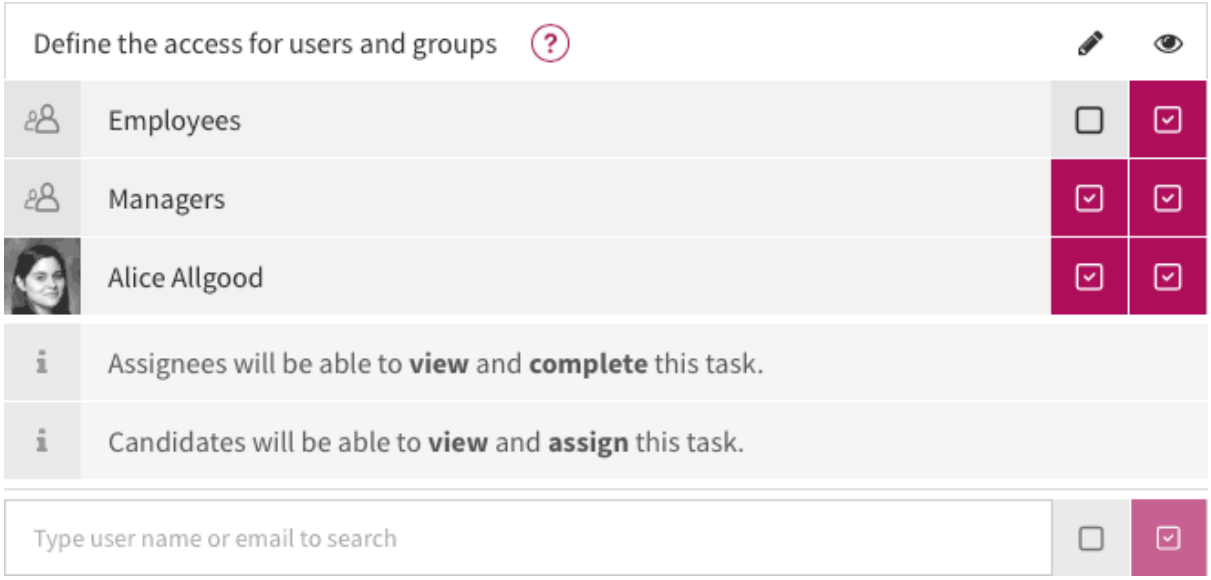
In the same way that you can restrict access to a whole process, you can also restrict access to individual user tasks in the process.

When you make a user task private, you can grant two different permissions to users and groups.

1. *View task* - review the task and participate in discussion by adding comments
2. *Edit task* - change the task's title, assignment and due date, and create subtasks.

Suppose you have a process that includes an approval, where someone from a *Managers* group must approve or reject a request from someone in the *Employees* group. You need to use the *Edit task* permission to restrict access to the approval user task, so that only managers can provide the approval.

To apply user task restrictions, open a process and select the user task. In the user task configuration panel, select the *Access Rights* tab. Click the *Define specific access* button to configure access control.



Configuring user task access control

You can now use the text box to search for users and groups, and use the checkboxes to grant permissions.

Chapter 11

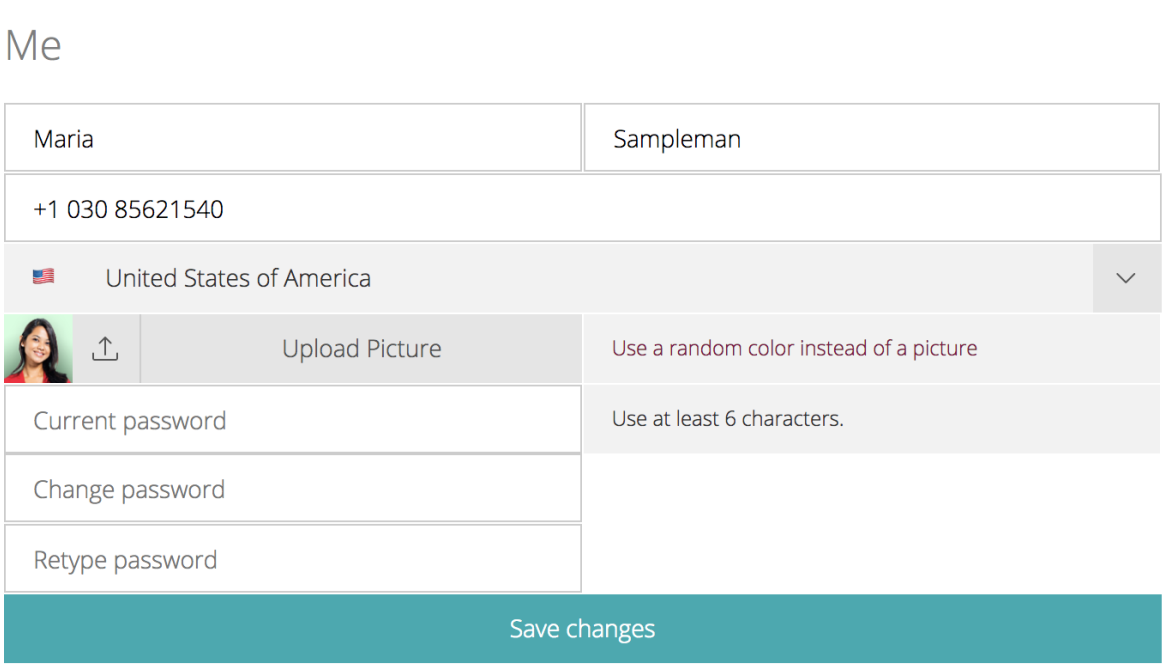
My profile

On the top-right drop-down menu, select *My Profile* to view and edit your own Signavio Workflow Accelerator user settings. The profile page has four sections:

- *Me* (page 110)
- *Preferences* (page 111)
- *Organizations* (page 111)
- *Services* (page 112)

11.1 Me

Use the *Me* section to update your user profile, including contact details and avatar image, or change your password.



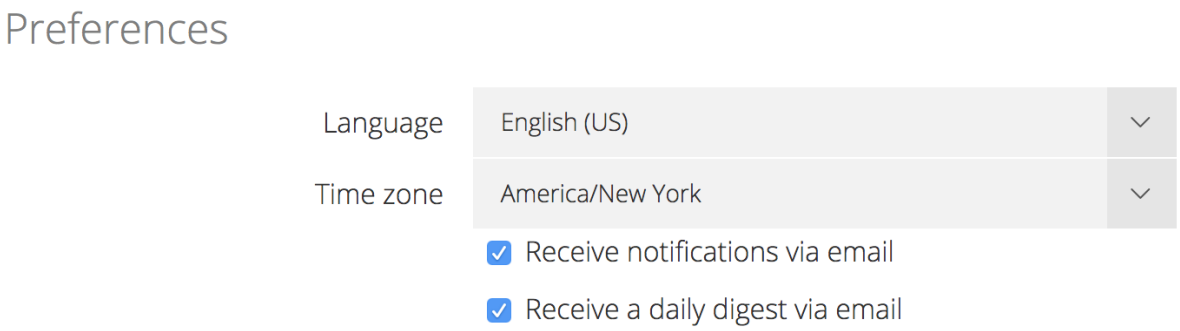
If you have not yet set a password, enter one in the 'Change password' field. Enter it again in the 'Retype password' field. Workflow Accelerator will check to ensure the new password meets security

requirements. When finished, click the 'Save changes' button.

To change your password, enter your current password into the 'Current password' field. In the next field, enter your desired new password. Enter it again in the 'Retype password' field. Workflow Accelerator will check to ensure the new password meets security requirements. When finished, click the 'Save changes' button.

11.2 Preferences

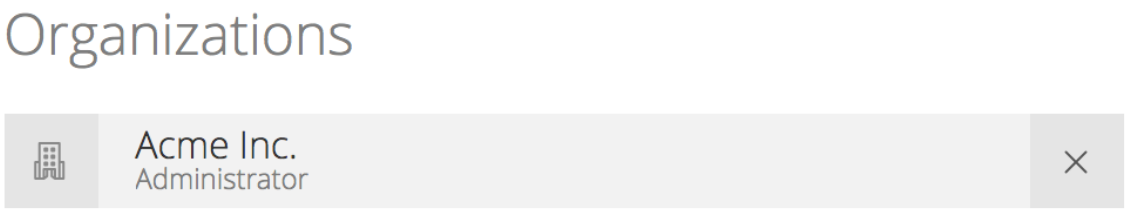
Use the *Preferences* section to configure the Workflow Accelerator user interface and notifications.



Uncheck the checkboxes to disable all email notifications.

11.3 Organizations

The *Organizations* section lists organizations that you belong to, and indicates whether you have the organization administrator role.

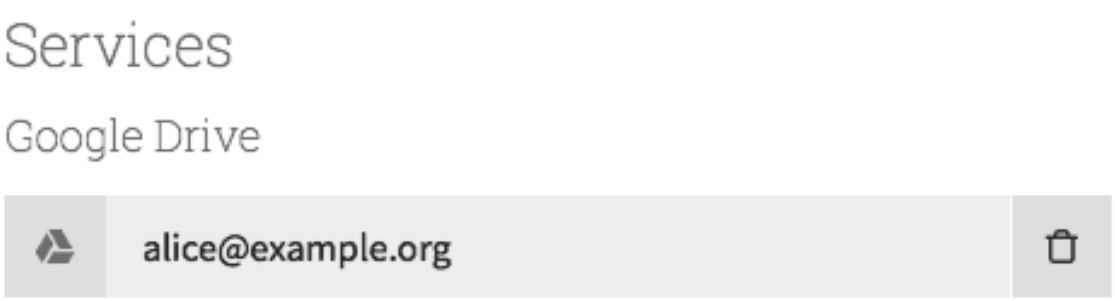


You can use the leave icon to leave an organization. When you leave an organization, Workflow Accelerator will:

- remove you from the organization's groups
- remove you from process models (process owner, access controls, action candidates)
- unassign you from tasks in the organization's processes
- remove your participation in cases of the organization's processes
- free up one of the organization's Workflow Accelerator licenses.

11.4 Services

The *Services* section shows your personal configurations for third-party services, such as a linked Google Account.



Chapter 12

Organization settings

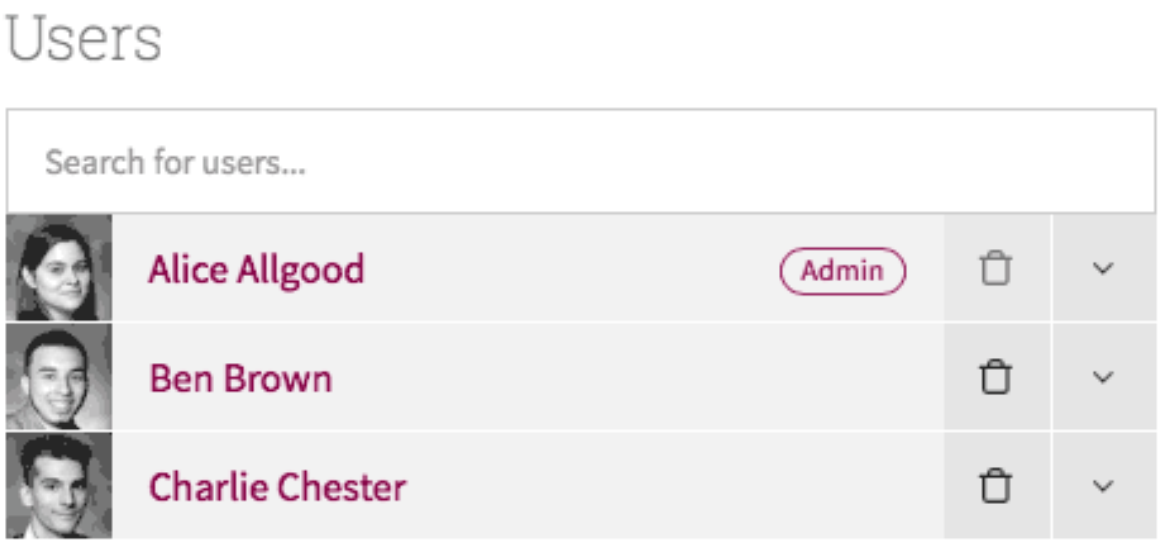
In Signavio Workflow Accelerator, your organization represents a collection of Workflow Accelerator users - typically a company - together with all their data in Workflow Accelerator. People outside your organization cannot see your organization's data. After you log into Workflow Accelerator, you see all the data inside one particular organization. If you belong to multiple organizations, you can switch between organizations by selecting a different organization under your name in the top right corner.

Use the Organization settings page to set-up users and groups, manage invitations, and configure external services such as *Salesforce Integration* (page 161).

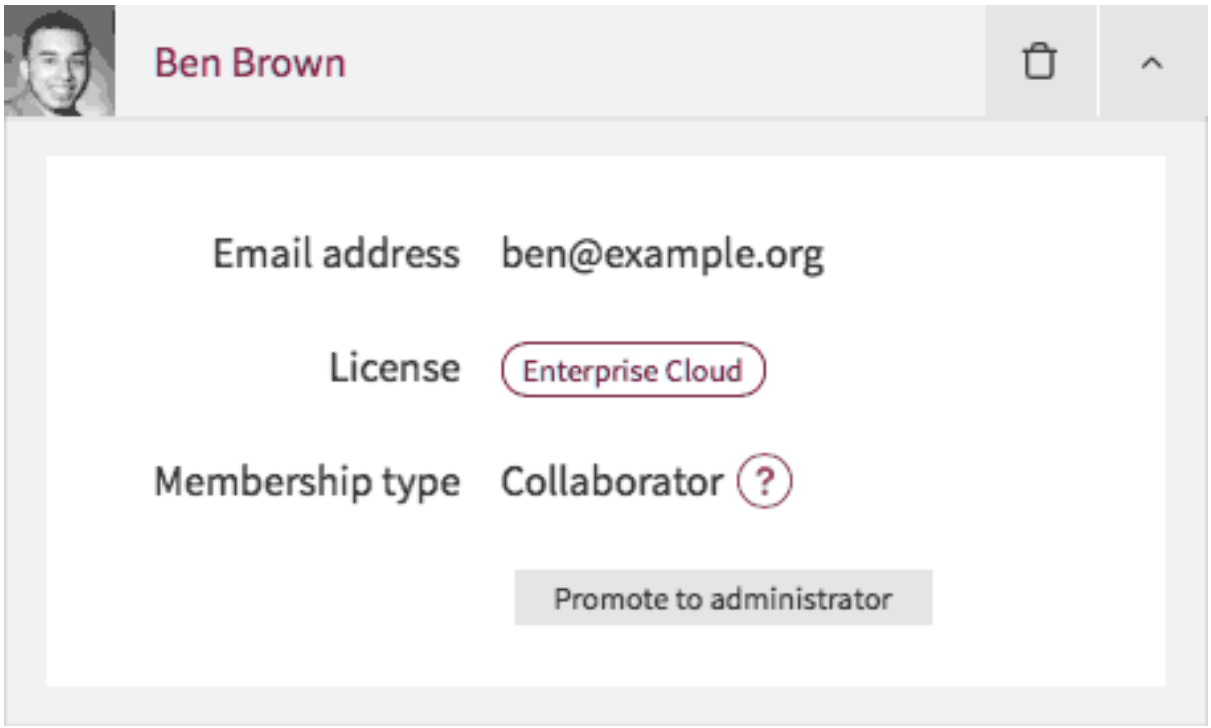
At the top of the page, in the masthead, you can edit the organization name.

12.1 Users

The users list shows users who belong to your organization. You can expand each user's entry to see their email address, license type, and membership type - indicating whether the user has the administrator role.



Administrators can promote other users to administrator: on an expanded user panel, click the *Promote to administrator* button.



12.2 Replacements

Administrators can also delete users, using the delete button next to the user name.

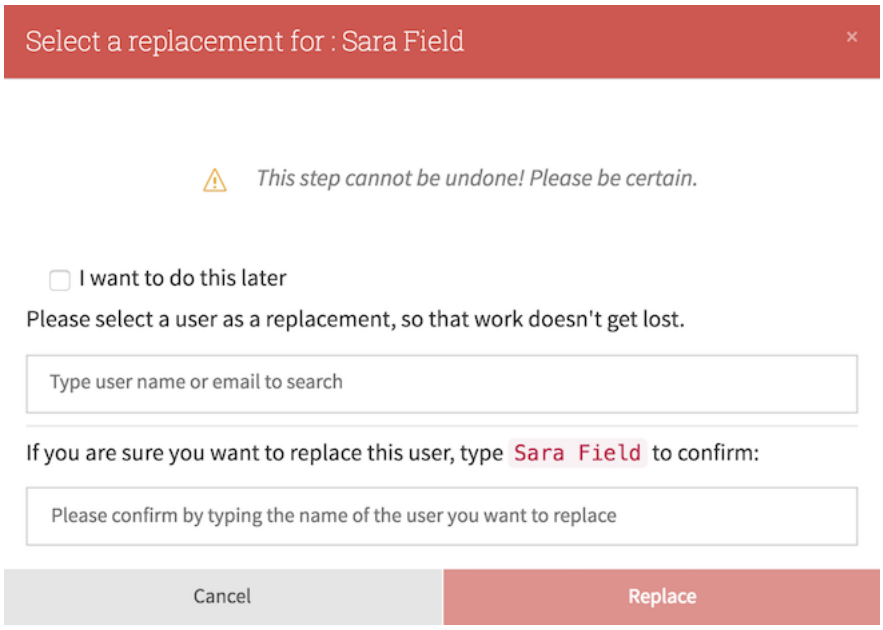
If you want to delete a user account, for example, because the user is leaving your organization, unfinished tasks to this account may still exist. In this case, specify a replacement who will take over open task assignments. The substitute will then have exactly all access rights to complete open tasks and cases, but not automatically inherits group memberships of the deleted user. Please also be aware that the assignment for closed tasks is not changed for audit reasons.

The substitute takes over

- in workflows: owner, assignments, candidates, default values for form fields, transition conditions, JavaScript test values, access rights
- in reports: owner, access rights
- in open tasks: assignments, candidates, access rights
- in cases: access rights

Hint: Deleted users are removed from all groups in which they were included. You must manually assign their replacement.

Whenever you delete a user, a dialogue where you can define a replacement appears.



Deleted users with task assignments and without replacement set, are listed as *Former users* and administrators are weekly reminded to specify a substitute.

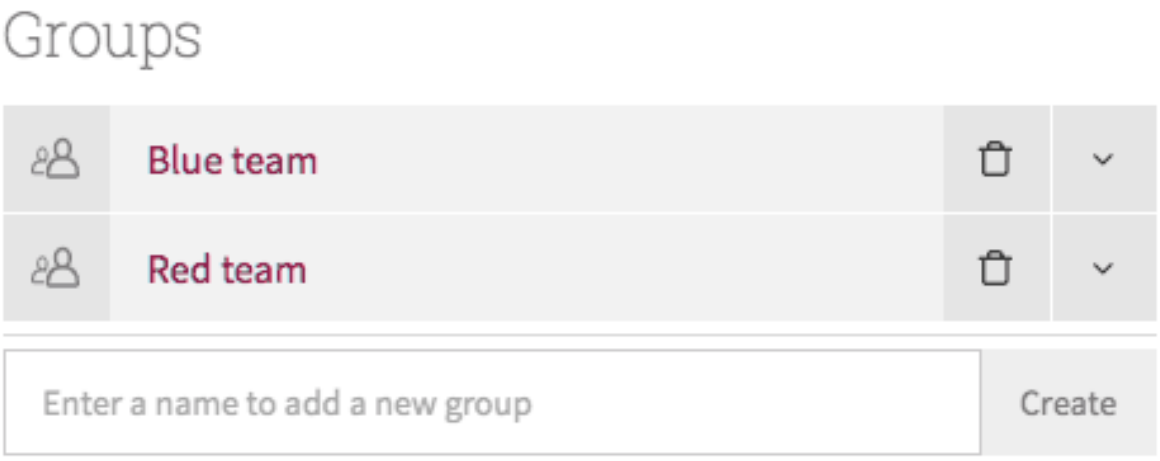
12.3 Invitations

The invitations list shows who you have invited to use Workflow Accelerator.

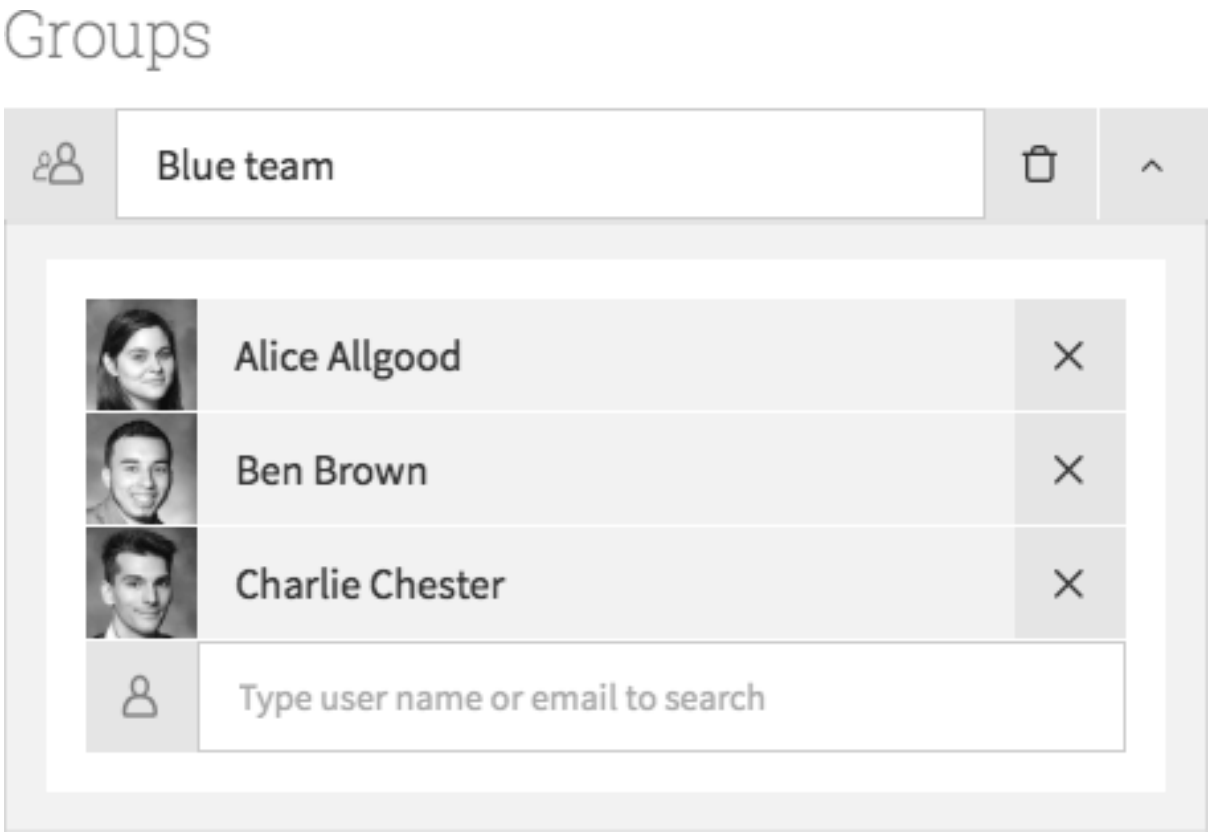
To invite someone to use Workflow Accelerator, select the license they will use, enter their email address in the text field and click the *Invite* button. They will receive an email with a link to the registration page, where they can create a Workflow Accelerator user that will become a member of the organization.

12.4 Groups

The groups list shows the organization's user groups. You can use these groups to define candidates for tasks in the *process builder* (page 28).



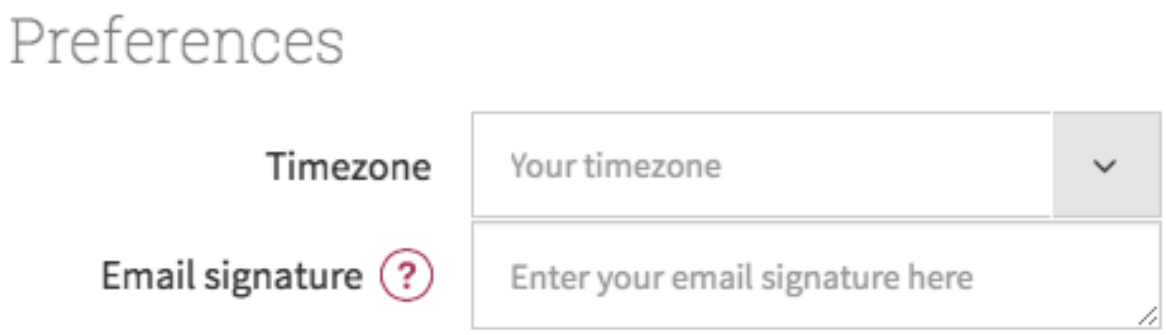
To create a new group, in the text field below the group list, enter a group name and click the *Create* button.



Click a group's name to expand its list of members, so you can edit the name and add or remove members.

12.5 Preferences

The *Preferences* include additional options that apply to the whole organization.

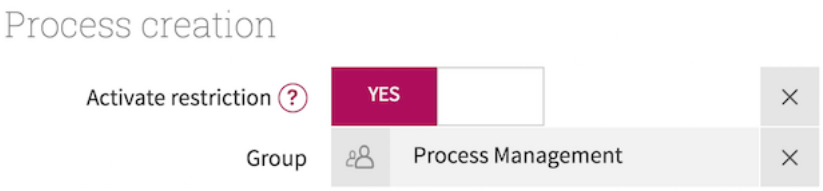


Time zone affects how Workflow Accelerator displays times.

Email signature replaces the default *Workflow Accelerator team* signature at the bottom of *notification emails* (page 120).

12.6 Process creation

New in version 3.34.



You can activate the *Process creation* option to restrict the right to create processes to a specific *user group* (page 115).

Only users of this group can

- create new processes,
- copy processes,
- import processes.

Users who are not members of the defined group, but have editing rights for specific processes are still able to modify these processes.

Note: The transfer of processes between Process Manager and Workflow Accelerator is not affected by this restriction. Any modeler can transfer a process from Process Manager to Workflow Accelerator.

12.7 Services

Use the *Services* tab to configure integration with cloud services for members of the organization to use. See *Salesforce Integration* (page 161).

12.8 Billing

Use the *Billing* tab to manage your organization's licenses for Workflow Accelerator. This page shows:

- the number of remaining user licenses - how many more people you can add to the organization
- the license expiry date, after which you must renew your licenses to continue using Workflow Accelerator.

The left-hand sidebar summarises your current license type. Click the *Upgrade your license* button to upgrade to another license type.

12.9 Single Sign-On

Single sign-on (SSO) makes it possible to access Workflow Accelerator using an existing corporate user account, so you do not have to log in to Workflow Accelerator separately. To request SSO for your organization, send a request using the *Send feedback* option in the application, including your SAML 2.0 Identity Provider Metadata[18].

[18] https://en.wikipedia.org/wiki/SAML_2.0#Identity_Provider_Metadata

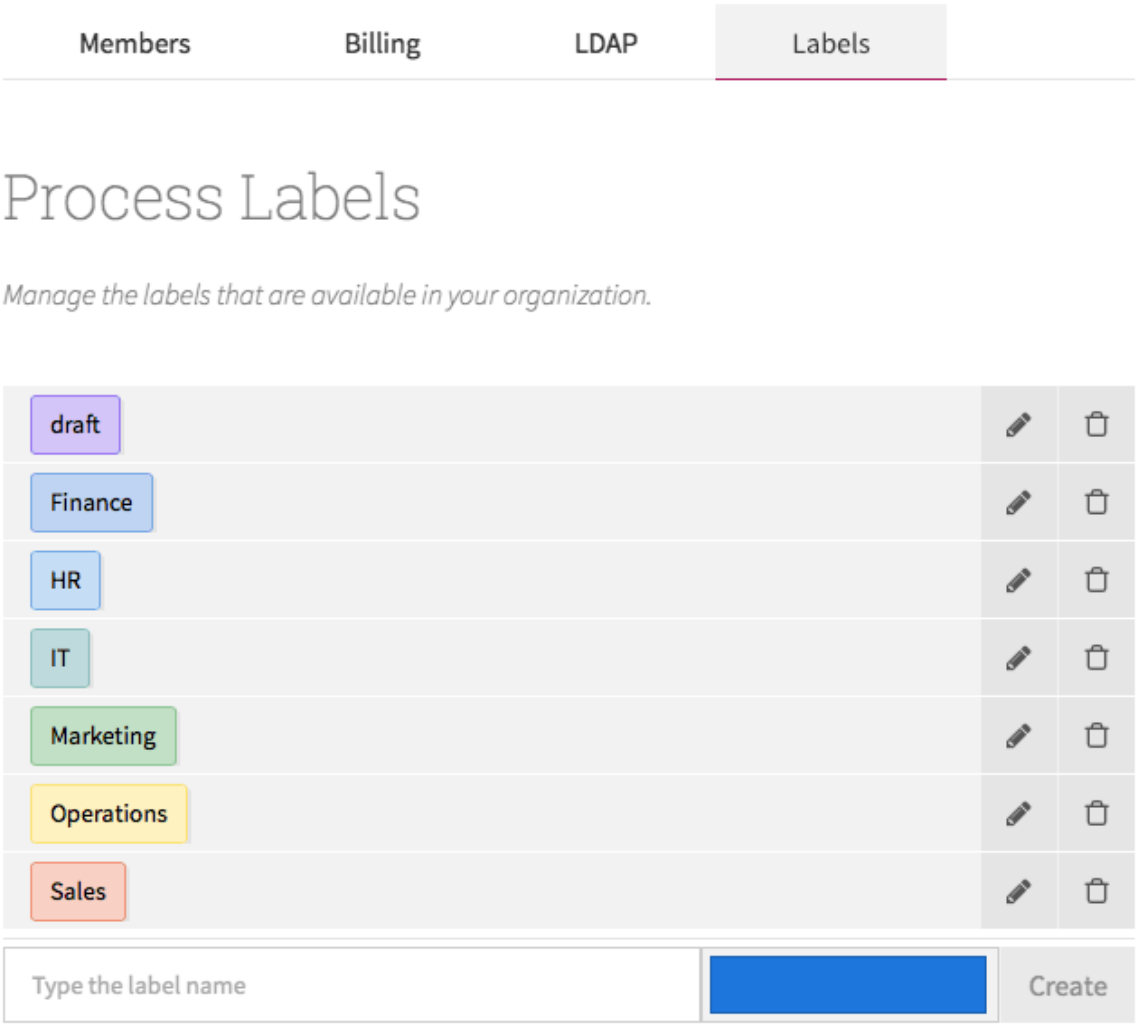
Workflow Accelerator currently only supports the G Suite[19] (formerly Google Apps) SSO provider.

Workflow Accelerator supports Security Assertion Markup Language (SAML) 2.0[20] Identity Provider-initiated SSO using the HTTP POST binding. Please contact us if you want to use a SAML 2.0 Identity Provider other than those listed above.

12.10 Labels

Most organizations soon have enough processes to make it inconvenient to browse the processes list. To keep your processes tidy, you can define and use labels, to categorize processes by department, status or however you like.

Use the *Labels* tab to define labels for your organization. You start with a set of default labels, but you can customize the list.



Configuring labels - used to categorize processes

To add a new label, enter a name in the text input field and select *Create*. Select a label or its edit icon to change its name or color. To delete a label, select the delete icon on the far right.

[19] https://support.google.com/a/answer/6087519?hl=en&ref_topic=6304963
[20] https://en.wikipedia.org/wiki/SAML_2.0

Chapter 13

Implementation guidelines

The implementation guidelines are available in the HTML version of the user guide at https://docs.signavio.com/userguide/workflow/en/guidelines.html.

Chapter 14

Notifications reference

Signavio Workflow Accelerator sends a variety of email notifications, to keep process participants up-to-date with cases they work on and to avoid task handover delays, when someone assigns a task.

To avoid unnecessary notifications, Workflow Accelerator waits a short time before sending notifications and omits notifications that have become obsolete.

14.1 Case due

This notification indicates that a case has reached its due date.

Workflow Accelerator sends this notification to the case's process owner.

14.2 Case task due

This notification indicates that a case has reached its due date and has open tasks.

Workflow Accelerator sends this notification to the assignees of open tasks in the case.

14.3 Task created

This notification indicates that the process has created a new task within a case.

Workflow Accelerator sends this notification to the task's default assignee, if the task has one. Workflow Accelerator also sends this notification to each of the task candidates, or every member of each candidate group, if defined.

Workflow Accelerator does not send this notification if you disable *Receive notifications via email* in *Preferences* (page 111).

14.4 Task assigned

This notification indicates that someone has assigned an existing task within a case.

Workflow Accelerator sends this notification to the task's new assignee.

Workflow Accelerator does not send this notification if you disable *Receive notifications via email* in *Preferences* (page 111).

14.5 Mentioned in a comment

This notification indicates that a comment on a case has mentioned someone.

Workflow Accelerator sends this notification to each user mentioned in the comment.

Workflow Accelerator does not send this notification if you disable *Receive notifications via email* in *Preferences* (page 111).

14.6 Reminder scheduled

This notification reminds case participants that a task remains open.

Workflow Accelerator sends this notification to the task's assignee when someone assigns the task, or to all of the task's candidates if the task does not have an assignee. If the task does not have an assignee or candidates, Workflow Accelerator sends the notification to the process' owner.

14.7 Task escalated

This notification indicates that an open task has reached its escalation deadline.

Workflow Accelerator sends this notification to the task's new assignee, as configured in the user task, or every member of each candidate group, if defined.

Workflow Accelerator always sends this notification, even if recipients have disabled *Receive notifications via email*.

14.8 New user registered

Workflow Accelerator sends a registration notification when someone registers a Workflow Accelerator trial account.

Workflow Accelerator sends this notification to the email address entered on the registration form.

14.9 Invited to join organization

This notification indicates that an organization administrator has invited someone to join an organization. Workflow Accelerator also sends a reminder for this notification when an administrator selects the option to resend the invitation.

Workflow Accelerator sends this notification to the person the administrator invited, who may or may not already have a Workflow Accelerator account.

14.10 Invitation cancelled

This notification informs someone that an organization administrator has cancelled an open invitation to join an organization.

Workflow Accelerator sends this notification to the person the administrator invited.

14.11 Invitation resent

This notification reminds someone that organization administrator has invited them to join an organization. Workspace administrators can trigger this notification manually.

Workflow Accelerator sends this notification to the person the administrator invited.

14.12 Password reset

Workflow Accelerator sends a password reset notification when someone uses the *Reset your password* option.

Workflow Accelerator sends this notification to the email address entered on the password reset form.

14.13 Service account access expired

This notification indicates that access to an external service, such as *Google Drive* (page 64), has expired.

Workflow Accelerator sends this notification to the Workflow Accelerator user who configured their external service account.

14.14 License about to expire

This notification indicates that a Workflow Accelerator license will soon expire.

Workflow Accelerator sends this notification to the administrators of the the Workflow Accelerator organization whose license will expire.

Chapter 15

Variables reference

See *Variables* (page 43) for an introduction.

15.1 Case variable

The *Case* variable contains the data that starting a case creates. This variable has several fields. You cannot change most of this data, except for the case name and due date.

An email has a composite type, with the following properties.

Property	JavaScript	Type	Description
Case ID	`id`	*ID* (page 126)	Unique identifier
Name	`name`	*Text* (page 126)	Entered or generated editable name
Case number	`caseNumber`	*Number* (page 126)	Sequential case number
Creator	`creatorId`	*User* (page 127)	User who created the case
Start date	`createTime`	*Date* (page 125)	Date and time the *Creator* started the case
Due date	`dueDate`	*Date* (page 125)	Optional editable due date
Priority	`priority`	*Text* (page 126)	Case priority - values `'0'` (high) to `'3'` (low)
Case link	`link`	*Text* (page 126)	URL of the case page in Signavio Workflow Accelerator
Cycle time	`duration`	*Duration* (page 125)	The duration the case has been open
Milestone	`milestone`	*Text* (page 126)	The last milestone the case has passed

15.1.1 Case ID

The case variable's *ID* uniquely identifies this case among cases for all processes.

15.1.2 Name

The *Name* field stores the name that either the *Creator* (page 124) entered manually, or that Workflow Accelerator generated. Case participants can edit this name. You might use the case name variable in a *Send email* (page 62) action configuration, to send emails that clearly identify their context by prefixing the subject line with the case name.

You can update the case name in a *JavaScript action* (page 141) by assigning a value to `_case.name`.

15.1.3 Case number

The *Number* field stores a sequential case number. Each process uses a separate case number sequence for its cases. In a *Handle customer order* process, you could use the case number as a generated order reference, for example.

15.1.4 Creator

The *Creator* field records the Workflow Accelerator user who started the case. The *User* (page 127) type includes name and email address fields, so you can use the case creator to configure a *Send email* (page 62) action that automatically notifies the requestor of an approval process' result.

15.1.5 Start date

The *Start date* records when the *Creator* (page 124) started the case. In an order process, for example, you could use this as the order date.

15.1.6 Due date

The *Due date* field stores the due date that case participants can set on the case view. Unlike the other case variables, the due date does not always have a value.

You can update the case due date in a *JavaScript action* (page 141) by assigning a value to `_case.dueDate`.

15.1.7 Case link

The *Link* field stores the URL of the case page in Workflow Accelerator. You can include this link in the body of email you send using a *Send email* (page 62) action, so that the recipient can immediately open the case in Workflow Accelerator. Workflow Accelerator includes this link its own built-in *email notifications* (page 120).

15.2 Trigger email variable

The *Trigger email* variable contains the email that started the case, for processes that have an *Email trigger* (page 33). You can use this variable to use information from the email that started the case during the process, and to send email to the sender.

This variable has the fields that the *Email* (page 125) type defines. You cannot change their values.

15.3 Data types

Variables store workflow information. Each variable has a user-defined name and a type. A type can represent a single value, like text or an email address. 'Composite' types such as user, file or email consist of several values.

Composite types have a list of fields, each with its own (possibly composite) type. The table of fields includes the JavaScript name that you use to access a field in JavaScript code.

When using expressions or configuring some action, you refer to workflow data. This can use references to variables, or to fields inside composite variables. The user interface guides you and shows the options you have.

15.3.1 Choice

The *Choice* type stores a text value from a fixed list of configured options.

15.3.2 Date

The *Date* type stores a date, or a date and a time of day.

15.3.3 Duration

The *Duration* type stores the length of a period of time, such as *2 weeks*. In the user interface, you can select from different time units. In JavaScript actions, *Duration* values store a number of seconds.

15.3.4 Email

The *email type* stores an email, which the email trigger uses. A variable stores the email that triggers the process. You can use the email's data fields, such as *from address* or the attachments, later in the process.

An email has a composite type, with the follow properties.

Property	JavaScript	Type	Description
ID	`id`	*ID* (page 126)	The unique identifier for this email
From	`from`	*Email address* (page 125)	The sender email address
From name	`fromName`	*Text* (page 126)	The sender's display name (optional)
To	`to`	*List* (page 126) of *Email address* (page 125)	The email addresses of the recipients
Reply to	`replyTo`	*Email address* (page 125)	The email address to send replies to (optional)
CC	`cc`	*List* (page 126) of *Email address* (page 125)	Email addresses that receive a copy of the message (optional)
Subject	`subject`	*Text* (page 126)	The subject of the email (optional)
Body text	`bodyText`	*Text* (page 126)	The plain text message (optional)
Body HTML	`bodyHtml`	*Text* (page 126)	The HTML code for an HTML email (optional)
Attachments	`attachmentIds`	*List* (page 126) of *File* (page 125)	The files to attach to the email (optional)

15.3.5 Email address

The *email address* type stores an email address.

15.3.6 File

The *file* type stores a reference to a file. JavaScript actions can use an API for *reading file contents* (page 140).

Property	JavaScript	Type	Description
ID	`id`	*ID* (page 126)	The unique identifier for this email
Content type	`contentType`	*Text* (page 126)	The file's media type[21]
File name	`name`	*Text* (page 126)	The file's name
Owner	`ownerId`	*User* (page 127)	The user who uploaded the file

15.3.7 ID

An ID has special kind of string type. Workflow Accelerator creates these IDs to identify objects; they have no other meaning. An ID string looks like `53fae958036471cea136ea83`.

15.3.8 Link

The *Link* type stores an Internet address (URL), such as a web site address.

15.3.9 List

A list simply represents a collection of values. The list has an order and all elements have a single specified type.

15.3.10 Money

The *Money* type stores a currency amount for a particular currency.

Property	JavaScript	Type	Description
Currency	`currency`	*Text* (page 126)	An ISO 4217[22] currency code
Amount	`amount`	*Number* (page 126)	The currency amount

15.3.11 Number

The *Number* type stores a number.

15.3.12 Object types

Objects have a composite type, with a list of named fields. For example: a user has an object type, with fields such as `firstName`, `lastName`, `mailAddress`.

15.3.13 Text

The *Text* type stores plain text.

[21] https://en.wikipedia.org/wiki/Media_type
[22] https://en.wikipedia.org/wiki/ISO_4217

15.3.14 User

A variable of type *user* refers to a user in your organization. A user has an object type, with the following properties.

Property	JavaScript	Type	Description
ID	`id`	*ID* (page 126)	The unique identifier for this user
Email address	`emailAddress`	*Email address* (page 125)	The user's email address
First name	`firstName`	*Text* (page 126)	The user's first name
Last name	`lastName`	*Text* (page 126)	The user's last name

15.3.15 Yes/No Checkbox

The *Yes/No Checkbox* type stores a single value that represents either *Yes* or *No*.

Chapter 16

Keyboard shortcuts

In some parts of the software, you can use the keyboard as well as the mouse for specific operations.

16.1 Process builder

arrow keys Move the selected diagram element

Delete Delete the selected diagram element

16.2 Case details view

R Refresh case

Chapter 17

Markdown

Markdown is a markup language which allows a text-to-html conversion. It was invented by John Gruber (http://daringfireball.net), mostly based on email formatting. You can use Markdown to format descriptions, comments, and even emails sent out from Workflow Accelerator.

The following is a brief overview of how to use Signavio's variation of Markdown.

17.1 Headers

Larger headers can be formatted using = or #. Sub-headings can be created using - or multiple #.

For example:

```
For a heading
===========
Use an '=' under your desired heading text

For sub-headings
-----------------------
Use dashes under the sub-heading text

# heading one
## heading two
### heading three
#### heading four
Use one up to four '#' and a space in front of the desired heading text
```

and the rendered result will appear like so:

For a heading
Use an '=' under your desired heading text

For sub-headings

Use dashes under the sub-heading text

heading one

heading two

heading three

heading four
Use one up to four '#' and a space in front of the desired heading text

17.2 Hyperlinks

Hyperlinks can be used in descriptions or comments.

In-text URLs (like `http://example.com`) are automatically turned into hyperlinks without the need for additional formatting.

If, however, you want to use a specific word or phrase as a hyperlink, use the following format: `[Word or phrase you want to turn into a link](http://example.com)`

17.3 Emphasis

For **bold** and *italic* text, use `**double**` and `*single asterisks*`, respectively.

17.4 Line breaks

Line breaks are empty lines used to separate blocks of code. Signavio does not support multiple empty lines.

In the editor, empty lines look like this:

```
This is a paragraph
with multiple lines

This is a new paragraph



Multiple empty lines between paragraphs
will be treated like a single empty line
```

and the rendered result will *always* look like this, no matter how many empty lines you use:

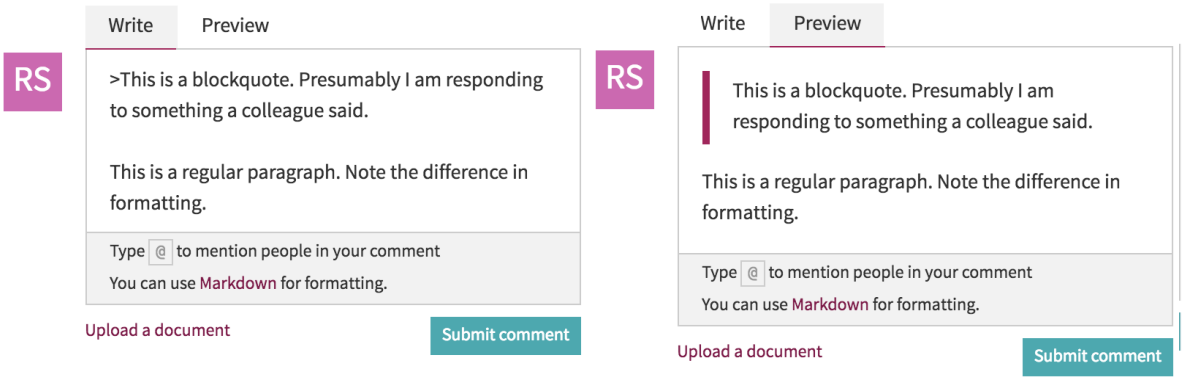
This is a paragraph
with multiple lines

This is a new paragraph

Multiple empty lines between paragraphs
will be treated like a single empty line

17.5 Blockquotes

Use > followed by a space to create blockquotes in your text. If you don't follow the > with a space, the formatting won't work.



17.6 Lists

You can easily create numbered or bulleted lists using Markdown.

For numbered lists, simply type it as you would normally.

For bulleted lists, use an * or - for each bullet, with a space between the asterisk or dash and the beginning of the bulleted point.

17.7 Horizontal rule

Use three or more ---, *** or ____ to create horizontal rules in your text.

17.8 Tables

You can create tables with Markdown, without having to copy and paste them from another application.

To create a table, use | to separate the different columns. Use at least 3 dashes - to separate the header cells from the table body, and use colons : to align the columns.

```
| Signavio | Markdown | Guide |
| ---------- | :----------: | -------: |
| column 1 | words | so |
| column 2 | are | are |
| column 3 | neat | tables |
```

When finished, your table will look like this:

Signavio	Markdown	Guide
column 1	words	so
column 2	are	are
column 3	neat	tables

Note that you don't have to keep things perfectly aligned. Your table will still be rendered correctly.

17.9 Embedding images

To embed images, use the following format: `![Alt text](https://example.com/image-url-here "Optional hint")`

`Alt text` will be shown when the image cannot be loaded. `"Optional hint"` is an optional text, use it to show more information when hovering over the image.

17.10 Inline code and code blocks

Use ` to highlight inline code.

Single backticks are used for highlighting a single word or phrase. Use three backticks to wrap whole code blocks. Note that highlighting the code block will also preserve the line breaks and indentation.

```
Emphasize a single `word` or `a whole phrase` with single backticks.
```
```
Or wrap whole blocks of code

which also preserve line breaks
```
```

Emphasize a single `word` or `a whole phrase` with single backticks.

```
Or wrap whole blocks of code

which also preserve line breaks
```
```

17.11 Inline HTML

Workflow Accelerator does not support inline HTML.

Chapter 18

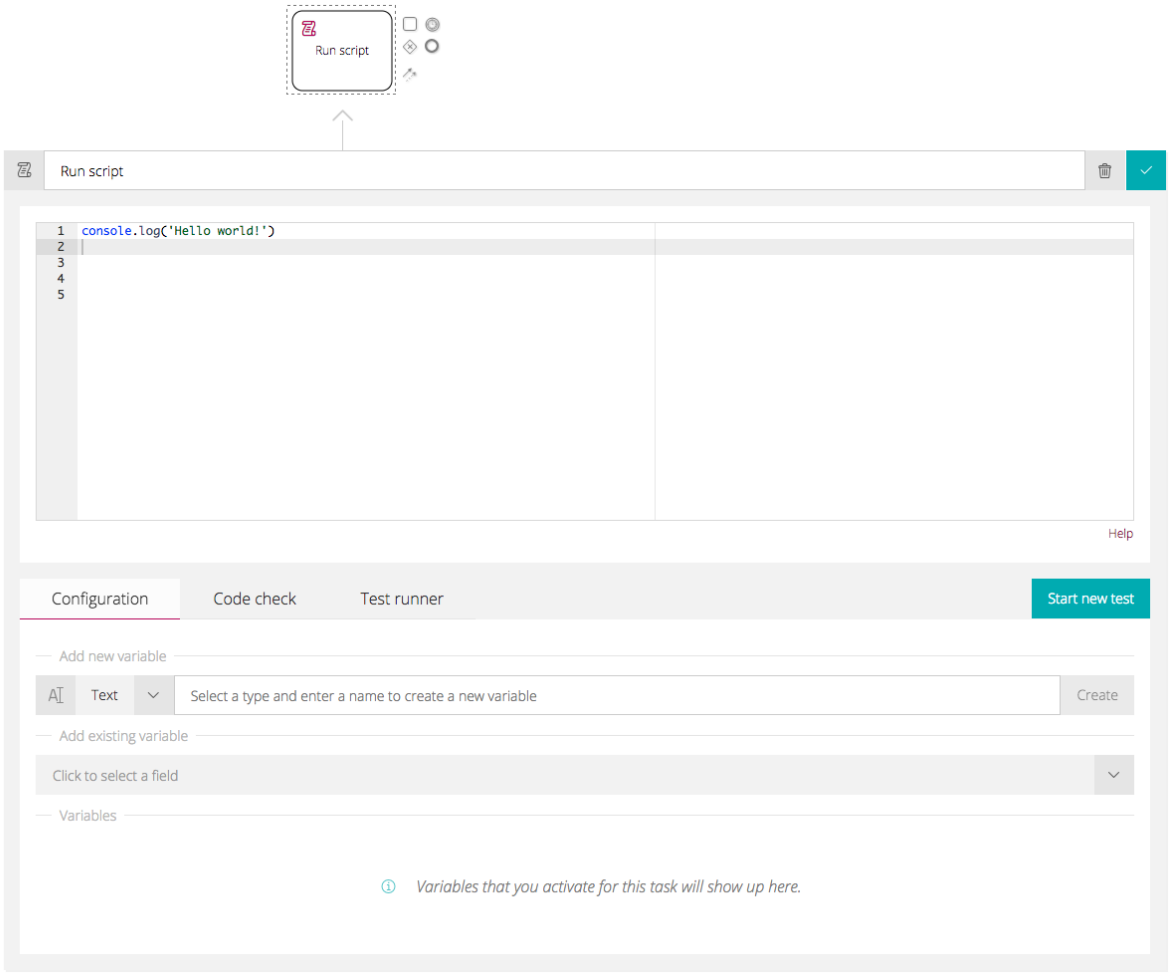
JavaScript Integration

You can integrate with external systems by writing JavaScript code in a *JavaScript action* (page 74). Signavio Workflow Accelerator runs the code on the server, using Node.js[23]. As well as Node.js' JavaScript API, scripts can use *additional libraries* (page 135).

18.1 JavaScript action configuration

After creating or selecting a JavaScript action, the configuration panel looks like this:

[23] https://nodejs.org



JavaScript configuration panel

The top section of the panel contains the JavaScript text editor. By default, it already contains `console.log('Hello World!');`. Use the console[24] API for log output when testing scripts.

18.2 JavaScript libraries

JavaScript actions support a number of popular JavaScript libraries. To import a package, use the `require` function:

```
var moment = require('moment');
```

You can also choose another name for the imported library:

```
var stringValidator = require('validator');
```

[24] https://nodejs.org/dist/latest-v5.x/docs/api/console.html

Supported JavaScript libraries

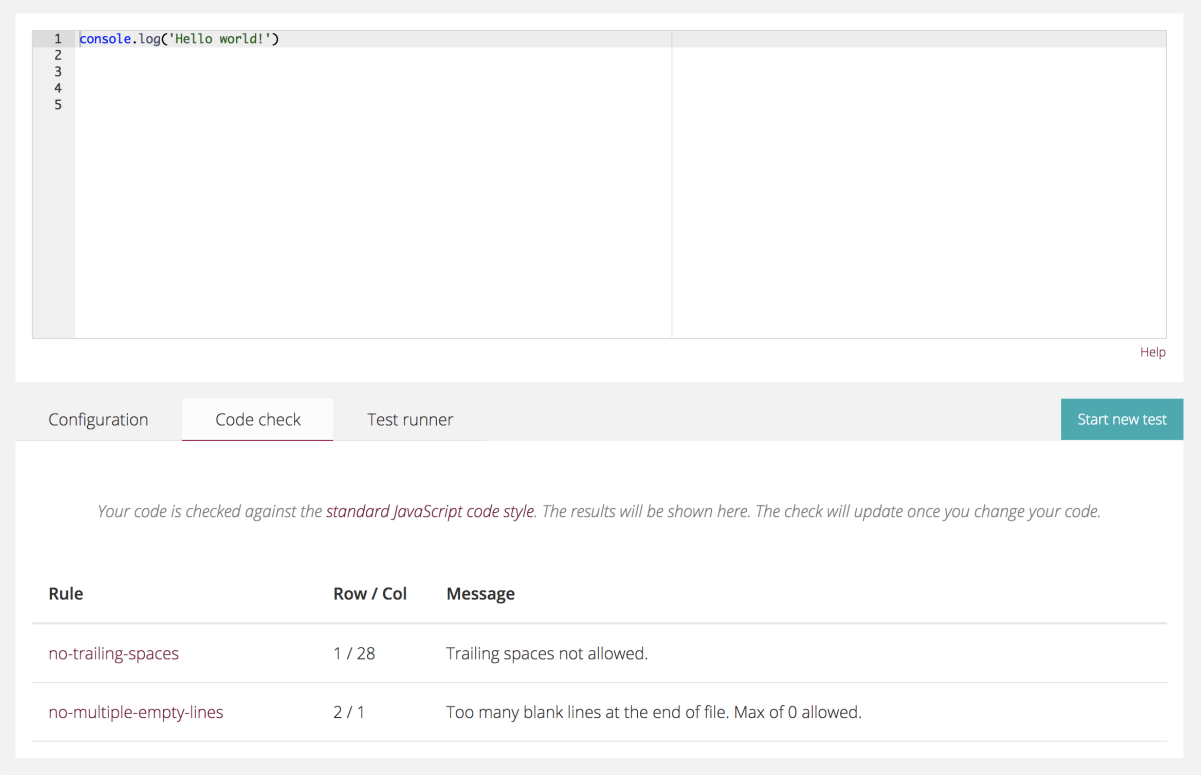
Library	Import	Description
CSV[25]	`csv`	CSV generation, parsing, transformation and serialization
Files	`files`	Built-in API for *File* (page 125) variable data
Lodash[26]	`lodash`	Convenience functions for working with collections and values
moment[27]	`moment`	Parse, validate, manipulate, and display dates; with Twix[28] date range, and moment-business-days[29] support
request[30]	`request`	Simplified HTTP request client
Users	`users`	Built-in API for *User* (page 127) variable data
validator[31]	`validator`	String validation and sanitization
WebDAV client[32]	`webdav-client`	Exchange files with a WebDAV endpoint
XML[33]	`xml-js`	XML generation and parsing

The JavaScript action always imports the `_` (Lodash) and `request` packages, for backwards compatibility.

18.3 Checking your code

Use the 'Code check' tab to check your code against the standard JavaScript code style[34]. Click 'Start new test' to execute your code (or simply click anywhere outside the panel). You will be brought to the 'Test runner' tab. Click the 'Code check' tab again to see the results.

[25] https://www.npmjs.com/package/csv
[26] https://www.npmjs.com/package/lodash
[27] https://www.npmjs.com/package/moment
[28] https://www.npmjs.com/package/twix
[29] https://github.com/kalmecak/moment-business-days
[30] https://www.npmjs.com/package/request
[31] https://www.npmjs.com/package/validator
[32] https://www.npmjs.com/package/webdav-client
[33] https://www.npmjs.com/package/xml-js
[34] https://standardjs.com/

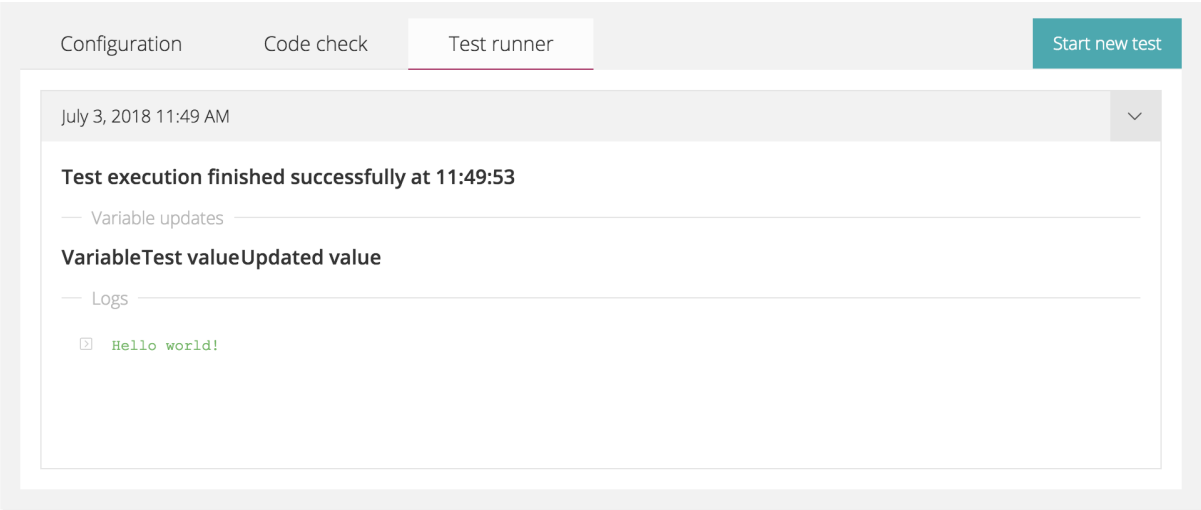


Code check results

The code check results indicate which, if any, JavaScript rule is violated, where the error occurs, and a message linking to https://standardjs.com/, where you can find more information about why it is an error.

18.4 Testing scripts

Use the *Test Runner* tab to test the script. Click *Start new test* to execute the JavaScript code. The test runner displays the results underneath:

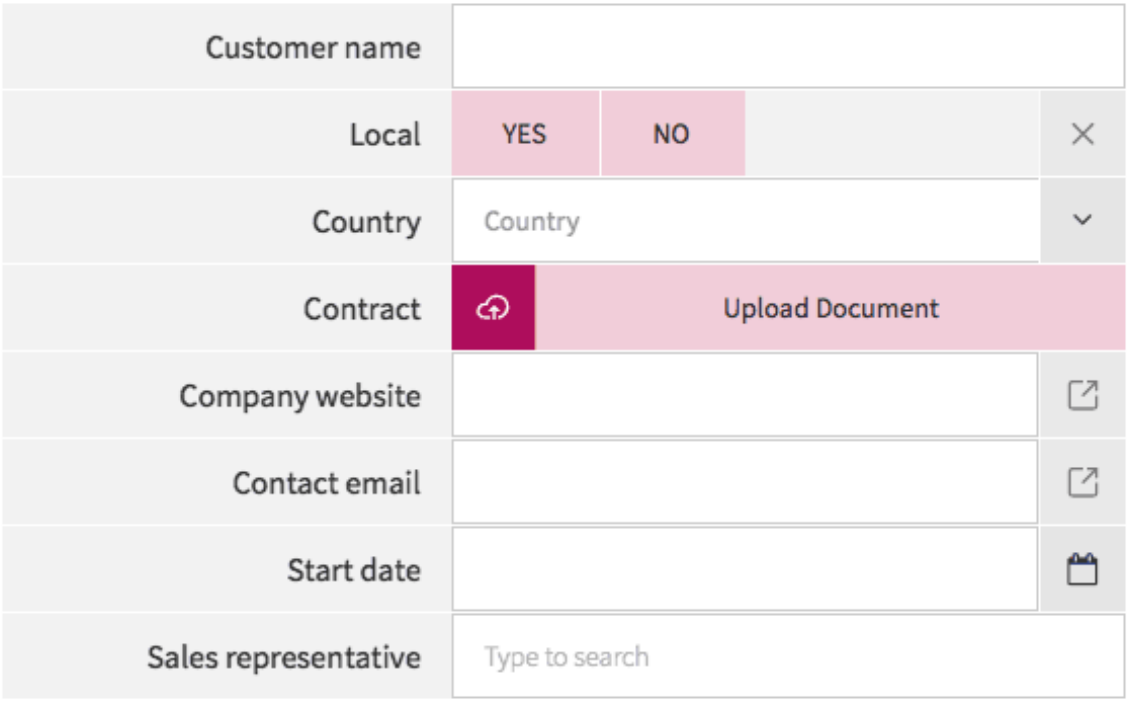


JavaScript test output

At the top, you see the test execution date and time. After running multiple tests, you can use this menu to select earlier test runs. The *Variable updates* section shows a table of *process variables* (page 43), with their test values and any updates. The *Logs* section shows console output and any errors.

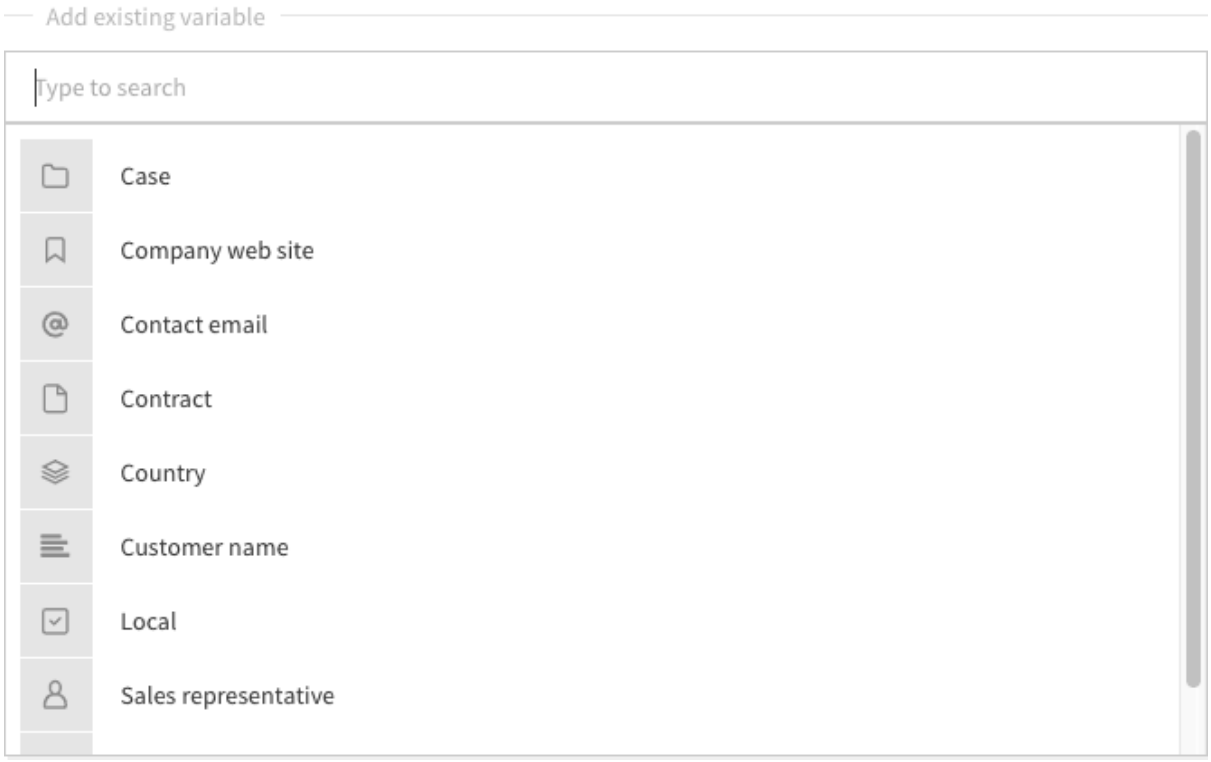
18.5 Using process variables

Next, we'll show how to work with data. Suppose that the process includes a form that has each type of field and looks like this:



Form fields that declare process variables

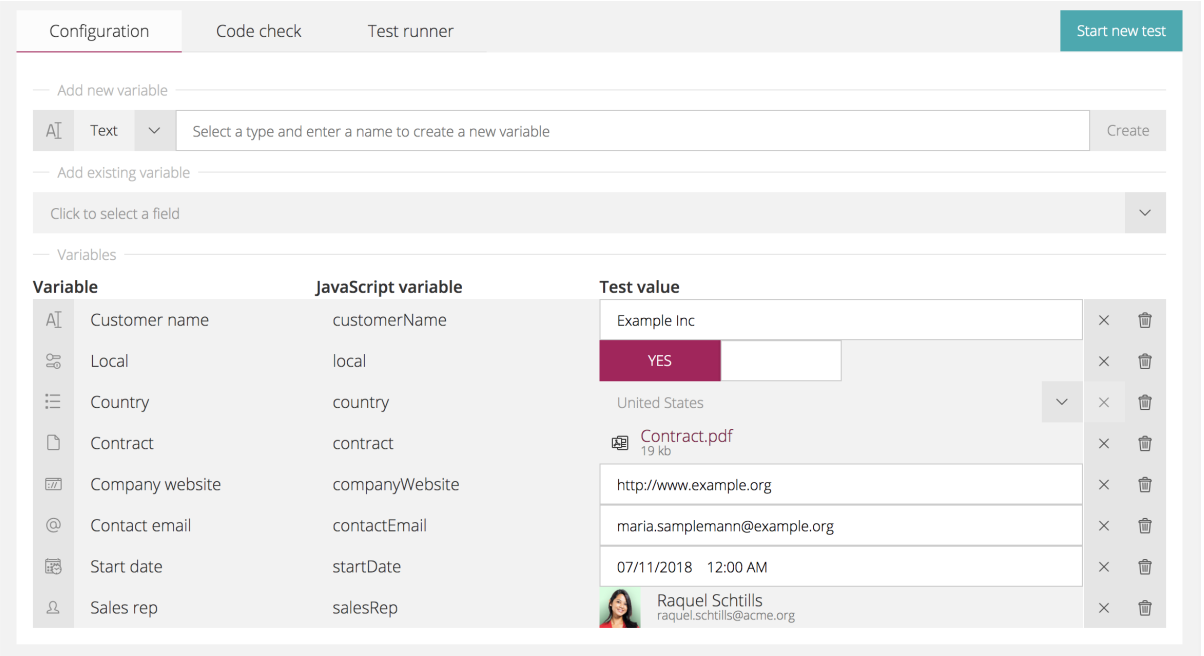
On the JavaScript configuration panel, the *Add existing variable* pick list now shows the form field variables.



Process variable selection

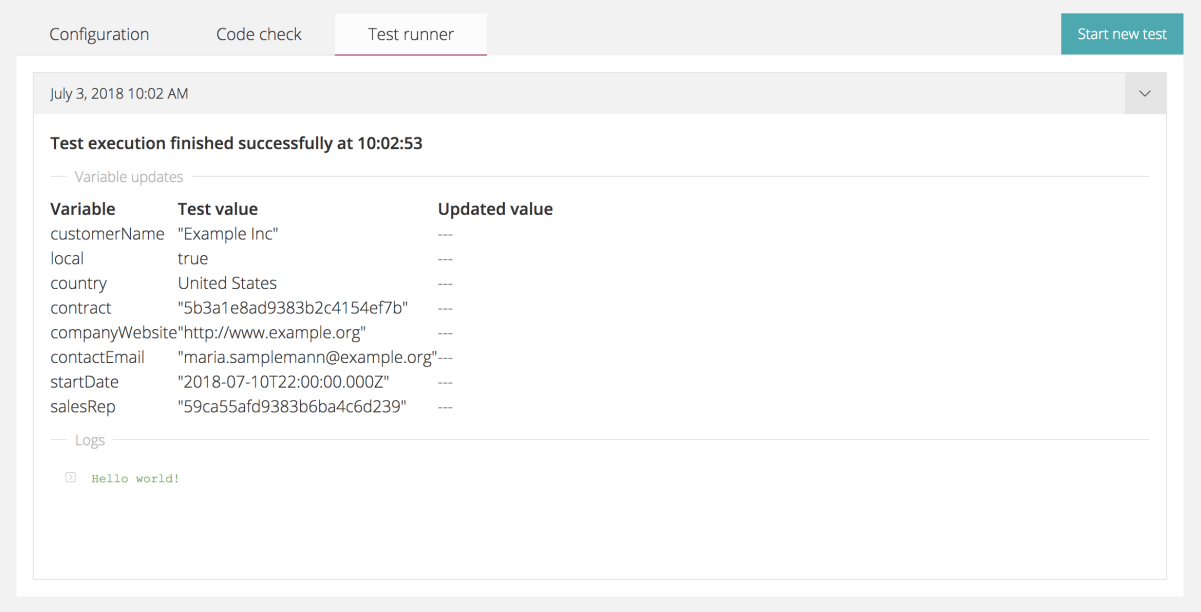
Select the variables you want to access in the script. The script can access the variables using the *JavaScript variable* name from the *Variables* table. To access object variables' fields, use the field names specified for the corresponding data type: *Case* (page 123), *Email* (page 125), *File* (page 125) or *User* (page 127).

In this example (below), you have selected all variables. For each variable that you select, you get an input field to specify a test value. Here you see all fields with a test value.



JavaScript test values

Clicking *Start new test* again to see the JSON structure of the variable data for the different variable types.



JavaScript JSON values

The *contract* and *salesRepresentative* variables have complex types, *File* (page 125) and *User* (page 127), so the table only shows an ID. The *Updated value* column shows the result of assigning new values to these variables in the script.

Note: You can use JavaScript actions to update process variables. Then, you need to make sure you *re-assign* a new value to the variable instead of mutating the variable itself. Otherwise, the system will ignore the update. For example, the system ignores `contactEmails.push('joan.doe@example.org')`, but correctly processes `contactEmails = [].concat([], 'joan.doe@example.org')`. This restriction doesn't apply to variables you only use in the context of the JavaScript action.

18.6 Reading file contents

A JavaScript action may need to read the contents of a file, in order to publish the file to an external web service. To access *file* (page 125) content, you need to require the `files` API.

```
const files = require('files')
const fileContent = files.getContent(contract)
```

In this example, `contract` is a file variable that references the file contents that the script reads.

The `getContent` function returns a Node.js File[35] object, whose `buffer` property provides access to the file content bytes. The following example loads a CSV file, converts the content bytes to a UTF-8 string, and parses the string:

```
const files = require('files')
const csv = require('csv')

// Read the reportCsv file variable
const csvFile = files.getContent(reportCsv.id)
```

[35] https://www.npmjs.com/package/file-api

```
csv.parse(csvFile.buffer.toString('utf-8'), {
    auto_parse: true,
    columns: true,
}, (error, data) => {
    console.log(data)
})
```

18.7 Updating case information

The process variables always include the built-in *Case* (page 123) variable, which contains information about the current case. Sometimes, you want to update this case information using data from process variables. You can update some of the this case variable's fields, as follows.

```
// Set the case name using a template.
_case.name = `Case ${_case.caseNumber}`;

// Set the case's due date using a date variable set on a form.
_case.dueDate = releaseDate;

// Set the case's priority, using text values '0' (high) to '3' (low)
// priorities defines constant values high, medium, normal, and low
const priorities = require('priorities')
_case.priority = priorities.low
```

A *case name template* (page 39) can only use *Form trigger* (page 31) fields to set the case name when the process starts. However, when you can set the case name directly in a JavaScript action, you don't have this restriction.

18.8 Loading user information

In a JavaScript action, you might need to select a Workflow Accelerator user based on external data, to assign a role. To do this, you can use the built-in `users` API to find a user by their email address.

```
const users = require('users');
reviewer = users.findByEmail(reviewerEmailAddress);
```

This example uses the value of a previously-supplied `reviewerEmailAddress` *Email address* (page 125) variable to set a `reviewer` *User* (page 127) variable.

18.9 Calling an external web service

You can use variables to send process data to an external web service, using the request module[36]. For example, the following script sends the value of the `startDate` variable in an HTTP POST request to an external web service.

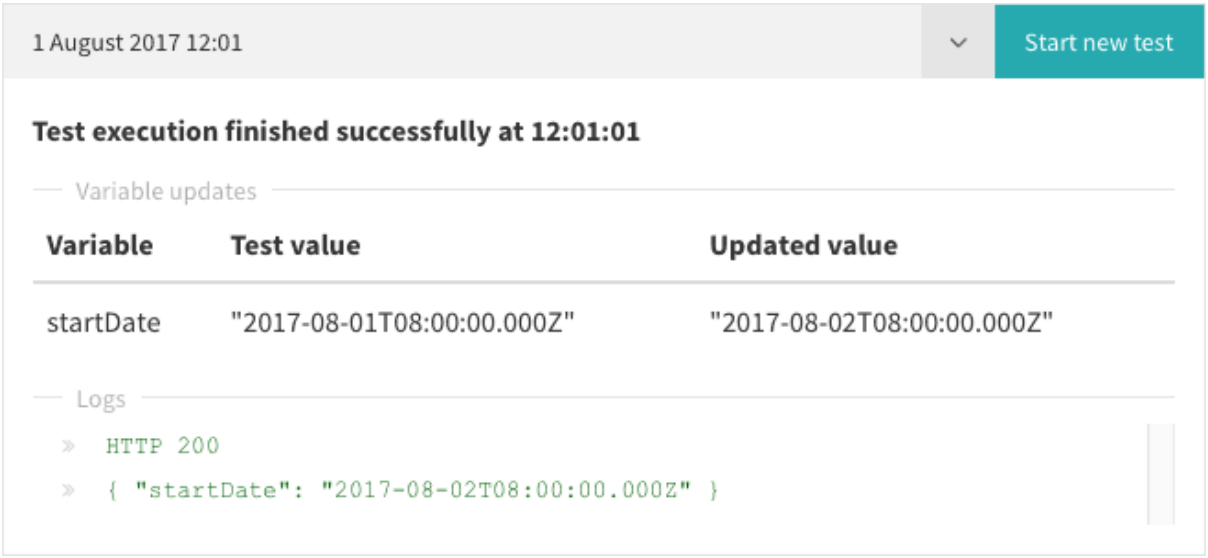
[36] https://github.com/mikeal/request/blob/master/README.md

```
const requestOptions = {
    url: 'http://www.mocky.io/v2/598050d7110000bf081cf957',
    body: JSON.stringify({ 'startDate': startDate }),
    headers: { 'Content-Type': 'application/json' }
}

const handleResponse = (error, response, responseBody) => {
    const log = `HTTP ${response.statusCode}\n${responseBody}`
    if (!error && response.statusCode == 200) {
        console.info(log)
        const responseData = JSON.parse(responseBody)
        startDate = responseData.startDate
    } else {
        console.error(log)
    }
}

request.post(requestOptions, handleResponse);
```

This example uses a test endpoint configured using Mocky[37] to return an HTTP response. This has the following result in the Workflow Accelerator test console:



Updating a variable via an external web service

The two log statements, starting with *HTTP 200*, show the HTTP response from the web service. The response body (as set-up in Mocky) contains JSON data that includes an updated value for the `startDate` variable, changing the date from `2017-08-01` to `2017-08-02`.

The script then parses this JSON response using `JSON.parse` and updates the `startDate` variable in Workflow Accelerator, as shown in the *Updated value* column in the test console's variables table.

18.10 Generating and parsing XML

XML is used as a common exchange format for data. In a JavaScript task you can parse and generate XML using the `xml-js` library.

[37] http://www.mocky.io/

The first example parses XML content of a local variable into a JavaScript object structure. You can also retrieve XML content from another source, like a process variable or an external web service.

The example uses the compact mode by setting the option `{ compact: true }` when calling the conversion function. The compact mode creates an object structure which resembles XML structure. It contains special properties like `_attributes` and `_text` which allow you to access the text content of XML elements and attributes.

```
const convert = require('xml-js')

const xml = `
<customers>
  <customer status="silver">John Doe</customer>
  <customer status="gold">Alice Allgood</customer>
</customers>
`

// Parse the XML string to a JavaScript object using the compact mode
const obj = convert.xml2js(xml, { compact : true})

const john = obj.customers.customer[0]
console.log(`${john._text} - ${john._attributes.status}`)
```

The second example shows how to generate XML from a JavaScript object. Note that the structure also contains the special properties `_attributes` and `_text` to indicate which parts of the object will be XML attributes or text content.

This example use compact mode to set the respective option.

```
const convert = require('xml-js')

const obj = {
    customers : {
        customer: [
            {
                _text: "John Doe",
                _attributes: {
                    status: "silver"
                }
            },
            {
                _text: "Alice Allgood",
                _attributes: {
                    status: "gold"
                }
            }
        ]
    }
}

// Generate the XML string from a JavaScript object
const xml = convert.js2xml(obj, { compact: true })

console.log(xml)
```

See the xml-js[38] library documentation for more information about generation and parsing options.

[38] https://github.com/nashwaan/xml-js#synopsis

18.11 Exchanging files with a WebDAV endpoint

In a JavaScript action, you might want to upload a file that has been uploaded to a task form to your own file server or load a file from an external server to use the content during the script execution. If your file server supports the WebDAV protocol, you can use the `webdav-client` library.

In this example, the content of the file variable `myFile` is uploaded to the endpoint `https://webdav.example.com`. When uploading the file content make sure to use the `buffer` property as the library expects either a buffer or a string for the file content.

```
const webdav = require('webdav-client')
const files = require('files')

// Create an authenticated connection to the WebDAV endpoint
const connectionOptions = {
  url:'https://webdav.example.com',
  authenticator: new webdav.BasicAuthenticator(),
  username:'signavio',
  password:'WyvsWGxWUtx3wTw1BkmfOtf4'
}
const connection = new webdav.Connection(connectionOptions)

// Read the content of the myFile file variable
const fileContent = files.getContent(myFile)

// Upload the content to the root directory using the original file name
connection.put(`/${myFile.name}`, fileContent.buffer, (error) => {
  if (error) {
    throw new Error(`File upload did not work: ${error}`)
  }
})
```

As an alternative, it is also possible to generate a string with the file content and upload it.

```
const webdav = require('webdav-client')

// Create a connection to the WebDAV endpoint
const connectionOptions = {
  url:'https://webdav.example.com',
  authenticator: new webdav.BasicAuthenticator(),
  username:'signavio',
  password:'WyvsWGxWUtx3wTw1BkmfOtf4'
}
const connection = new webdav.Connection(connectionOptions)

// Create the file content
const myNotes = 'The answer is 42!'

// Upload the content to the file myNotes.txt in directory /path/to
connection.put('/path/to/myNotes.txt', myNotes, (error) => {
  if (error) {
    throw new Error(`File upload did not work: ${error}`)
  }
})
```

Reading a file via WebDAV is as simple as uploading one. Instead of `put` use `get` and specify the desired file. The callback offers the second parameter `body` which contains the file content.

```
const webdav = require('webdav-client')

// Create a connection to the WebDAV endpoint
const connectionOptions = {
```

```
  url:'https://webdav.example.com',
  authenticator: new webdav.BasicAuthenticator(),
  username:'signavio',
  password:'WyvsWGxWUtx3wTw1BkmfOtf4'
}
const connection = new webdav.Connection(connectionOptions)

connection.get(`/foobar.txt`, (error, body) => {
  if (error) {
      throw new Error(`File upload did not work: ${error}`)
  }
  console.log(body)
})
```

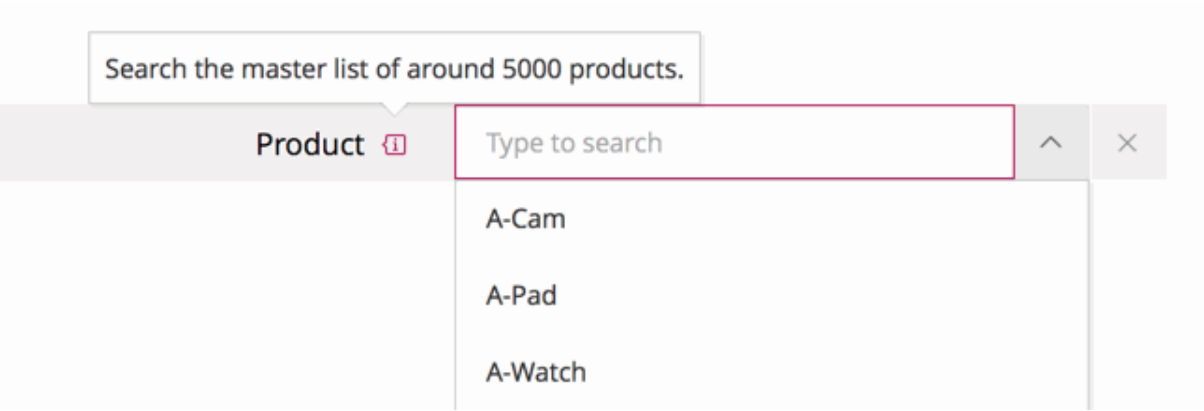
See the WebDAV[39] library documentation for more methods to modify your files, and additional request options.

[39] https://github.com/OpenMarshal/npm-WebDAV-Client#usage

Chapter 19

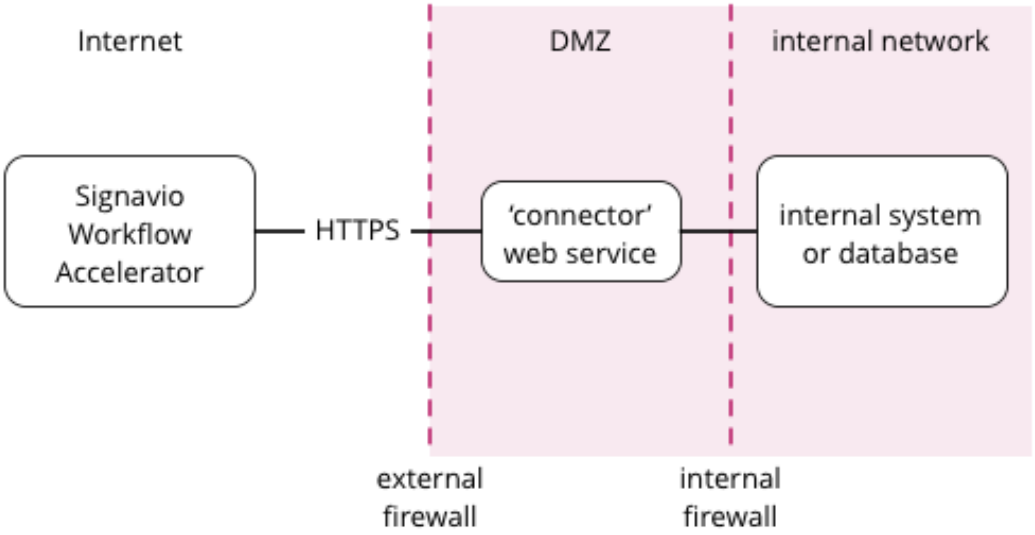
Custom data connectors

When you define a process in Signavio Workflow Accelerator, you often include your own data in the process definition, such as the list of options for a form field. This works well for small lists that don't change often or that belong to the process, such as a list of document statuses in a document approval process. However, fixed lists in the process definition become difficult to maintain when the data changes frequently or includes a large number of items, such as a list of products or customers.



Use a custom data connector to populate choice options from an external database

With Workflow Accelerator, you can also integrate dynamic structured data from other IT systems into your workflows. The workflow system fetches data from a third-party system using a *connector*, which you implement and host yourself.



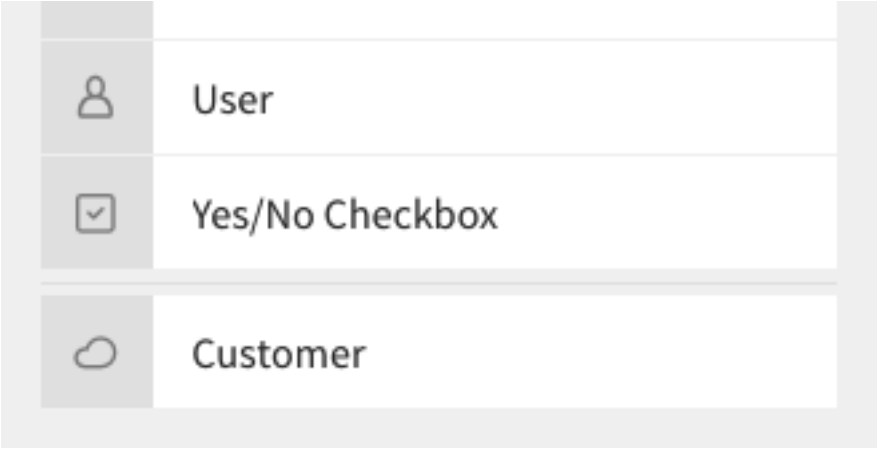
Connector deployment scenario for connecting to a corporate internal system

This diagram shows a typical scenario, with a connector that provides data from an internal database to Workflow Accelerator (SaaS), via the Internet. The connector web service runs in your corporate network's demilitarized zone[40] (DMZ), which your IT department configures to provide more security than allowing Workflow Accelerator to connect directly to your internal network.

A connector provides a web service that translates between the external system and Workflow Accelerator. The connector implements a defined interface, which Workflow Accelerator uses to access data in a format it can use. Workflow Accelerator and the connector communicate over *HTTP* or *HTTPS*, which makes it possible to implement connectors in any programming language.

19.1 Using a connector

A connector can provide data to *User task* (page 56) form fields. For example, you can create a connector that provides a list of customers, which adds a *Customer* type in the form builder:



A Customer *connector type in the form builder, at the bottom of the list of field types*

A connector reference field:

- makes it possible to select from a dynamic list of records

[40] https://en.wikipedia.org/wiki/DMZ_(computing)

- supports auto-complete so you can work with a large number of records
- can include structured data for each record.

19.2 Implementing a connector

To implement a connector, you make data available via four URLs, such as:

```
https://example.org/connector

https://example.org/connector/customer/options?filter=ACME

https://example.org/connector/customer/options/42

https://example.org/connector/customer/42
```

These four URLs correspond to four different kinds of resource.

1. *Connector descriptor* (page 149) - defines one or more record types, each of which defines a list of fields.
2. *Record type options* (page 151) - a list of records for each record type the connector defines.
3. *Record type option (single option)* (page 152) - a single record from the *Record type options* (page 151) list.
4. *Record details* (page 152) (optional) - all fields for one record from the list of records.

Workflow Accelerator accesses the connector on the web, via the public Internet, or via a private intranet for an on-premise installation. Workflow Accelerator calls the connector's URL the *endpoint URL*.

For example, consider a connector that accesses a fictional customer database, that you publish at the endpoint URL `https://example.org/connector`. In this example, each customer record has the following fields.

Example - customer record fields

Property	Description
`id`	Unique identifier
`name`	Full name
`email`	Email address
`subscriptionType`	Type of subscription - bronze, silver or gold
`discount`	Default customer discount
`since`	Registration date

A complete example customer record, formatted as *JSON*, would then look like this:

```
{
  "id" : "7g8h9i",
  "name" : "Charlie Chester",
  "email" : "charlie@example.org",
  "subscriptionType" : "silver",
  "discount" : 15,
  "since" : "2012-02-14T09:20:00.000Z"
}
```

This example now includes enough information to implement a complete connector.

19.2.1 Connector descriptor

A connector needs a descriptor to provide basic information, such as its name and description, as well as detailed information about the structure of the data the connector provides. When you implement a connector, you must make the descriptor available as the following HTTP resource.

URL / - the connector's *endpoint URL*

Request methods GET - fetches the connector descriptor

Response content type `application/json`

Response body A JSON object with the following fields.

Connector descriptor properties

Property	Description
`key`	Unique alphanumeric key (characters a-z, A-Z, 0-9) that identifies the connector
`name`	The connector name shown in the user interface
`description`	Detailed connector description
`typeDescriptors`	List of one or more descriptors for record types
`version`	The connector version, which should increase if the provided data structure changes
`protocolVersion`	The connector protocol version, currently `1`.

For example, the JSON response body for a connector descriptor without any type descriptors would look like this:

```
{
  "key" : "customers",
  "name" : "Customers",
  "description" : "A database with all customers.",
  "typeDescriptors" : [ ],
  "version" : 1,
  "protocolVersion" : 1
}
```

In our example, you would retrieve the connector descriptor by sending the HTTP request `GET https://example.org/connector/`. A **record type descriptor** describes the format of the data the connector provides, such as the format of a customer record. In the JSON response, the `typeDescriptors` property's value contains an array of record type descriptor JSON objects.

Record type descriptor properties

Property	Description
`key`	Unique alphanumeric key (characters a-z, A-Z, 0-9) that identifies the record type within the connector descriptor, used in *Record type options* (page 151) and *Record details* (page 152) URLs
`name`	The type name shown in the form builder user interface
`recordType`	Either `reference` or `value` - specifies how Workflow Accelerator stores selected values, see *Record storage* (page 153)
`fields`	An array of *record field descriptors* (page 150)
`optionsAvailable`	Boolean value - `true` indicates that the connector provides a list of record options, used to provide a list in the user interface for user selection
`fetchOneAvailable`	Boolean value - `true` indicates that Workflow Accelerator can fetch single records by the ID used in the options list

For example, the JSON object for a customer record type descriptor, without any fields, would look like this:

```
    {
      "key" : "customer",
      "name" : "Customer",
"recordType" : "value",
      "fields" : [ ],
      "optionsAvailable" : true,
      "fetchOneAvailable" : true
    }
```

A **record field descriptor** specifies one field of a record type. A record type has a complex structure that includes one or more fields, such as a customer's full name. Each field has a key, a name and a data type.

Record field descriptor properties

Property	Description
`key`	Unique alphanumeric key (characters a-z, A-Z, 0-9) that identifies the field type within the record type
`name`	The field name shown in the user interface
`type`	A JSON object that describes field's data type - see *Data types and formats* (page 153)

Hint: Every record type automatically includes `name` (when storing records by value) and `id` fields with type `text`, so you don't have to define them explicitly.

An example for the `email` of our customer record type looks like this

```
{
  "key" : "email",
  "name" : "Email",
  "type" : {
    "name" : "emailAddress"
  }
}
```

A complete example of our connector descriptor would look like this:

```
  {
    "key" : "customers",
    "name" : "Customers",
    "description" : "A database with all customers.",
    "typeDescriptors" : [ {
      "key" : "customer",
      "name" : "Customer",
"recordType" : "value",
      "fields" : [ {
        "key" : "email",
        "name" : "Email",
        "type" : {
          "name" : "emailAddress"
        }
      }, {
        "key" : "subscriptionType",
        "name" : "Type of the subscription",
        "type" : {
          "name" : "choice",
          "options" : [
                  {
                          "id" : "bronze",
                          "name" : "Bronze"
                  }, {
```

```
                        "id" : "silver",
                        "name" : "Silver"
                }, {
                        "id" : "gold",
                        "name" : "Gold"
                }
          ]
        }
      }, {
        "key" : "discount",
        "name" : "Discount",
        "type" : {
          "name" : "number"
        }
      }, {
        "key" : "since",
        "name" : "Registration date",
        "type" : {
          "name" : "date",
          "kind" : "datetime"
        }
      } ],
      "optionsAvailable" : true,
      "fetchOneAvailable" : true
    } ],
    "version" : 1,
    "protocolVersion" : 1
}
```

19.2.2 Record type options

When you use a record type on a form, you will see a form field where you can enter a search query and select one of the options shown. Each result represents a record provided by the connector. In order to show a selection of different records to the user, a connector can provide a list of options for a record type.

To make a list of options available to forms, in the *Connector descriptor* (page 149), set the `optionsAvailable` flag to `true`. The connector must also make the options available as the following HTTP resource.

URL (relative to the endpoint URL) `/:type/options` - with path parameter `:type` (a record type key)

Query string (optional) `filter=:query` - added when the user enters a search; `:query` encodes the search string

Request methods GET - fetches the list of record type options

Response content type `application/json`

Response body An array of JSON objects, which should have a limited maximum length. Each object in the array must have the following fields.

Record type options object properties

Property	Description
`id`	Unique string record ID
`name`	The text label shown in the user interface, which could aggregate multiple record fields like `name (email)`

For example, a list of customer options, with URL `https://example.org/connector/customer/options`, would look like this:

```
[ {
  "id" : "1a2b3c",
  "name" : "Alice Allgood"
}, {
  "id" : "4d5e6f",
  "name" : "Ben Brown"
}, {
  "id" : "7g8h9i",
  "name" : "Charlie Chester"
} ]
```

19.2.3 Record type option (single option)

After someone selects an option, the case user interface may later display the selected option in other contexts. Connectors that set the `optionsAvailable` flag to `true` must also make it possible to look up a single option by its ID, in order to display the option name.

URL (relative to the endpoint URL) `/:type/options/:id` - with path parameters `:type` (a record type key) and `:id` (the option ID)

Request methods GET - fetches a single record type option

Response content type `application/json`

Response body A single JSON object, with the same fields as the objects in the *Record type options* (page 151) response.

For example, a single customer option, with URL `https://example.org/connector/customer/options/1a2b3c`, would look like this:

```
{
  "id" : "1a2b3c",
  "name" : "Alice Allgood"
}
```

19.2.4 Record details

When you use a connector form field to select a record, you can use the record's data in the workflow. Depending on *Record storage* (page 153), Workflow Accelerator stores the whole record or only the record's ID as a reference, and fetches the entire record either directly or when needed, when accessing the nested data.

To make a record's fields available, in the *Connector descriptor* (page 149), set the `fetchOneAvailable` flag to `true`. The connector must also make the records available as the following HTTP resource.

URL (relative to the endpoint URL) `/:type/:id` - with path parameters `:type` - a record type key, and `:id` - a record ID

Request methods GET - fetches details for a single record

Response content type `application/json`

Response body A JSON object containing all fields of the record with the requested ID.

For example, a customer record, with URL `https://example.org/connector/customer/7g8h9i`, would look like this:

```
{
  "id" : "7g8h9i",
  "name" : "Charlie Chester",
  "email" : "charlie@example.org",
```

```
  "subscriptionType" : "silver",
  "discount" : 15,
  "since" : "2012-02-14T09:20:00.000Z"
}
```

Selecting this customer record from the customer options list would give the workflow access to all of this customer's fields. As well as the custom fields, there are three pre-defined properties, shown in the following table.

Record details properties

Property	Description
`id`	The text label shown in the user interface (required)
`name`	The text label shown in the user interface (required when storing records by value)
`version`	A string indicating the version of the record (optional)

19.3 Record storage

Workflow Accelerator supports two different ways of storing selected records: by reference or by value. The following table compares the two options.

Storing records by *value*	Storing records by *reference*
Recommended option for most use cases	Default option, for backwards compatibility with older connector implementations
Stores the entire record in the case	Stores the record ID in the case
Reads the whole record once, when selected	Read the record each time it is displayed
Cases, tasks and reports load faster	Cases, tasks and reports load slower
The case shows the record as it was when selected	The case shows the record's latest value
You can use connector types in reports	You cannot use record fields in filters, grouping or aggregation

The connector specifies each record type's storage option in its record type descriptor's `recordType` property. If the connector offers multiple types, they can use a mixture of *value* and *reference* options.

When you store records by value, the connector must always include a `name` field in the record data. Workflow Accelerator uses this name to display record values.

19.4 Data types and formats

A data type defines which kind of value and format a field in a record can have. A type descriptor represents a data type as a JSON object, whose `name` property contains the data type name.

Data types may use additional properties for type-specific configuration. Furthermore, the expected format of a record value depends on the data type.

19.4.1 Choice type

A choice type represents a value from a fixed list of configured options.

Property	Values
`name`	`choice`
`options`	A JSON array of choice option objects, each with `id` and `name` properties

The `id` property stores a unique alphanumeric key (characters a-z, A-Z, 0-9) that identifies the option within the choice type; no two options may have the same `id`. The user interface shows the `name` property's value to the user.

The data type JSON object for a choice type with three options looks like this:

```
"type" : {
  "name" : "choice",
  "options" : [
    {
      "id" : "b",
      "name" : "Bronze"
    },
    {
      "id" : "s",
      "name" : "Silver"
    },
    {
      "id" : "g",
      "name" : "Gold"
    }
  ]
}
```

A field value stores the `id` property's value:

```
"value" : "g"
```

19.4.2 Date type

A date represents either a date and time (such as *2012-02-14 09:20*), just a date (*2012-02-14*), or just a time (*09:20*).

Property	Values
`name`	`date`
`kind`	`date`, `time`, `datetime`; specifies whether the value describes a date, a time of day or both (required)

```
"type" : {
  "name" : "date",
  "kind" : "datetime"
}
```

Date values must always use the `YYYY-MM-DDThh:mm:ss.SSSZ` ISO 8601[41] date format and the UTC time zone. For example:

```
"value" : "2012-02-14T09:20:00.000Z"
```

All date types use this format - `datetime`, `date` and `time`. For `date` and `time` values, execution only uses the first and last parts of the values, respectively.

19.4.3 Email address type

An email address type represents an email address:

```
"type" : {
  "name" : "emailAddress"
}
```

[41] https://en.wikipedia.org/wiki/ISO_8601

An email address value stores a plain string:

```
"value" : "alice@example.org"
```

19.4.4 Link type

A link type represents an Internet address (a URL), such as a web site address:

```
"type" : {
  "name" : "link"
}
```

A link value stores a plain string:

```
"value" : "http://www.example.org/"
```

19.4.5 List type

A list type represents an ordered list of values of one of the other data types:

```
"type" : {
  "name" : "list",
  "elementType" : {
    "name" : "emailAddress"
  }
}
```

A list value stores an array of values:

```
"value" : [
  "alice@example.org",
  "bob@example.org",
  "charlie@example.org"
]
```

19.4.6 Money type

A money type represents the combination of an *amount* and a *currency*:

```
"type" : {
  "name" : "money"
}
```

A money value stores a JSON object with the fields `amount` and `currency`. The `amount` property stores a number. The `currency` property stores an ISO 4217[42] currency code:

```
"value" : {
  "amount" : 12.40
  "currency" : "EUR"
}
```

[42] https://en.wikipedia.org/wiki/ISO_4217

19.4.7 Number type

A number type represents either an integer or decimal number:

```
"type" : {
  "name" : "number"
}
```

A number value stores a plain number, using a single `.` as decimal separator:

```
{
  "integerValue" : 42,
  "decimalValue" : 42.42
}
```

19.4.8 Text type

A text type represents a string - either a single line of text or multiple lines. Optionally, to indicate that text may contain multiple lines, add the flag `multiLine` to the data type.

Property	Values
`name`	`text`
`multiLine`	(optional) if set to `true` the text field will allow multiple lines of input

```
"type" : {
  "name" : "text"
}
```

```
"type" : {
  "name" : "text",
  "multiLine" : true
}
```

A text value stores a plain string:

```
"value" : "Example"
```

19.4.9 Yes/No Checkbox type

A yes/no checkbox type represents a choice between the values 'yes' and 'no':

```
"type" : {
  "name" : "boolean"
}
```

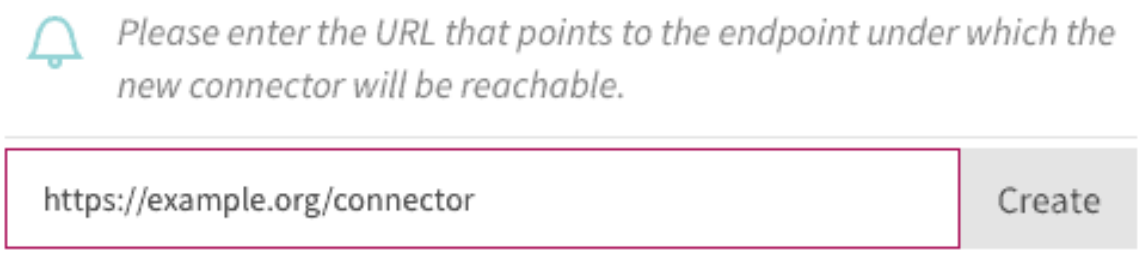
A yes/no checkbox value stores a Boolean value - `true` or `false`:

```
"value" : true
```

19.5 Configuring a connector

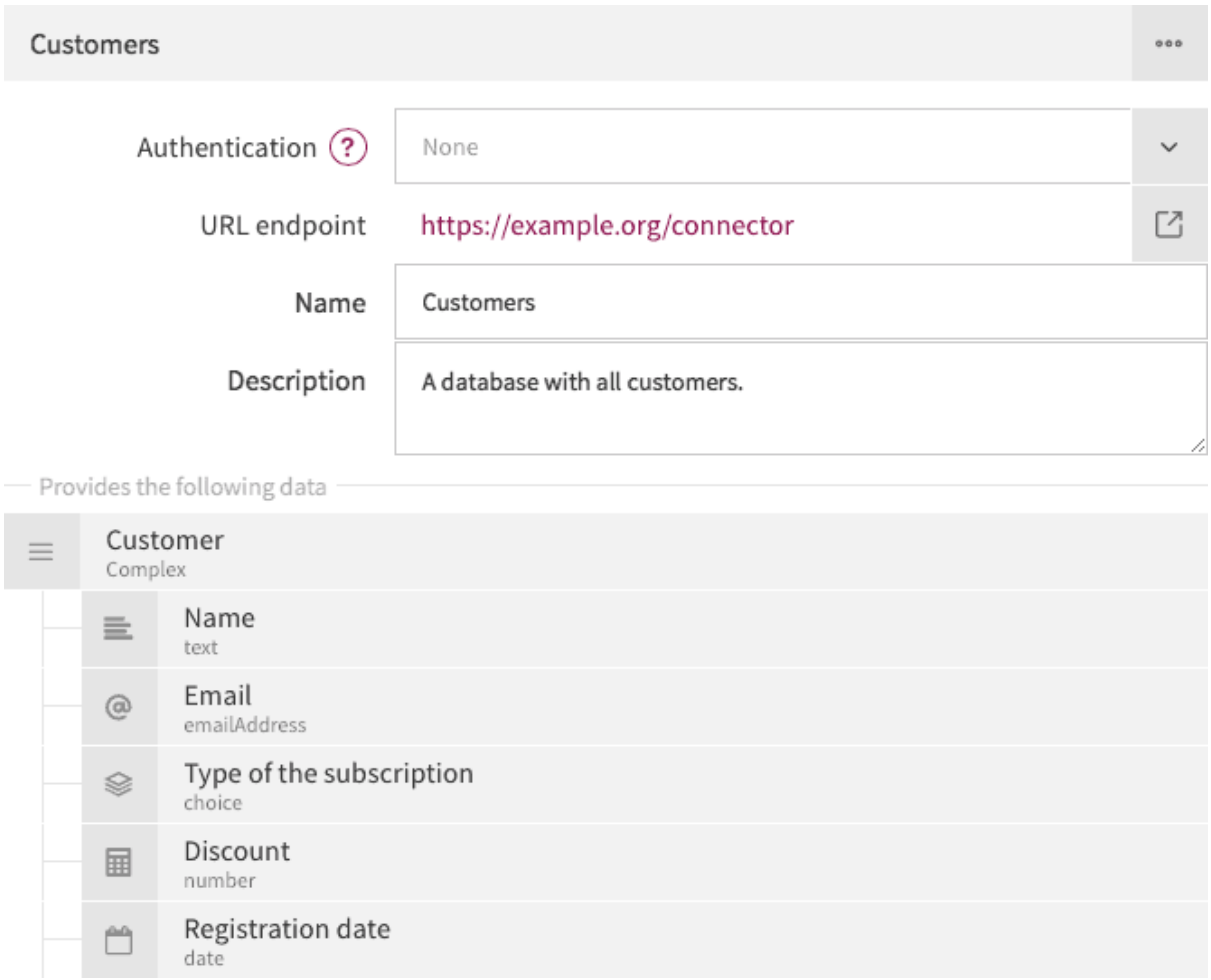
To configure connectors, on the top-right menu, select *Services & Connectors*, then select the *Connectors* tab. When you have published your connector, you can add it here.

Select *Add new connector* and enter the connector's endpoint URL.



Adding a new connector with the endpoint URL `https://example.org/connector`

When you add a connector, Workflow Accelerator fetches the connector descriptor and shows a summary:



Connector summary, including an overview of record and field types

If you make changes to your connector, such as adding or renaming a field, you need to reload the configuration. On the connector's top-right menu, select *Reload connector* to fetch the latest version of the descriptor.

19.6 Deleting a connector

You can delete a connector if you no longer wish to use it. On the connector's top-right menu, select *Delete connector* to remove its configuration from Workflow Accelerator. If you delete a connector by mistake, select *Add new connector* and enter the endpoint URL again.

19.7 Authentication

Publishing a connector makes it publicly accessible, as well as any data that the connector provides. To prevent unauthorized access, the connector can implement authentication, so that only Workflow Accelerator can access the data. Connectors may use one of two authentication mechanisms.

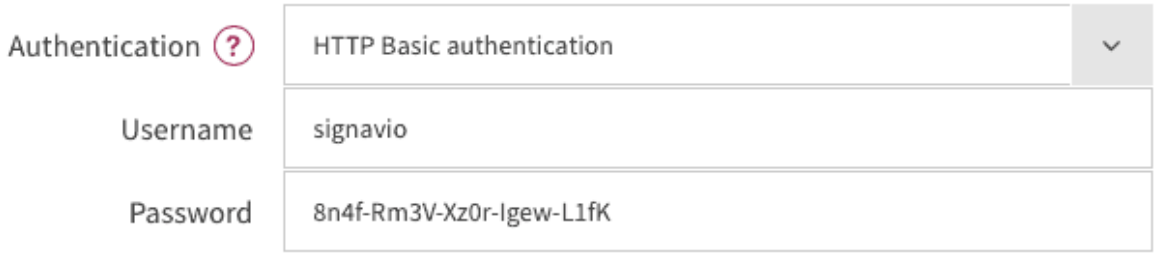
19.7.1 HTTP Basic authentication

Connectors can use HTTP basic authentication[43] to restrict access using a user name and password that you specify when configuring the connector. To implement HTTP Basic authentication, your connector endpoints must:

1. send an HTTP *401 Unauthorized* response, with an empty response body, for any request that does not include valid credentials
2. check the credentials in the `Authorization` HTTP header field, when provided, by decoding the Base64-encoded user name and password and verifying their values.

Warning: HTTP Basic authentication sends an unencrypted password over the network, so you should only allow access to private connectors via HTTPS.

To use basic authentication, use the *Authentication* field to select *HTTP Basic authentication*, and enter a user name and password:



Configuring basic authentication

When you configure a connector to use Basic authentication, Workflow Accelerator will pre-emptively include an `Authorization` header when sending requests to the connector endpoints. In Basic authentication, the header value consists of the authentication scheme name `Basic` followed by a space and the Base64-encoded user name and password, separated by a colon (`signavio:8n4f-Rm3V-Xz0r-Igew-L1fK`). This results in a request header that looks like:

```
Authorization: Basic c2lnbmF2aW86OG40Zi1SbTNWLVh6MHItSWdldy1MMWZL
```

Sending this header with every request avoids an additional *401 Unauthorized* response and a new request for the authentication challenge.

[43] https://en.wikipedia.org/wiki/Basic_access_authentication

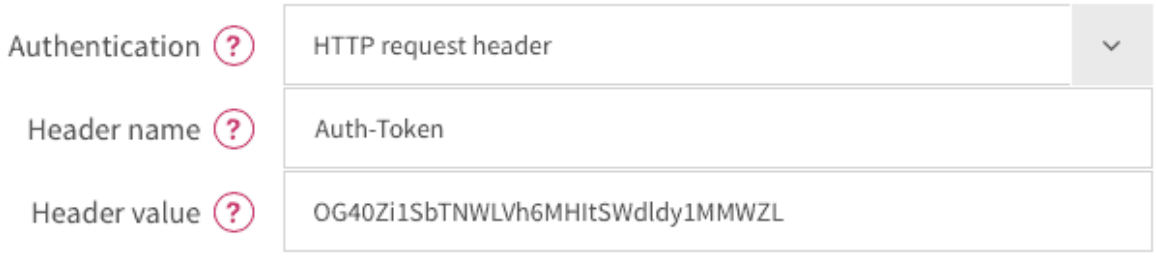
19.7.2 Token authentication

Similar to an API key, you can choose a password (token) that Workflow Accelerator will include in a request header field or URL query string, for every request it sends to the connector endpoints. In the connector configuration, you can choose between a request header field or a URL query string parameter, and specify the header or parameter name.

Warning: Token authentication sends an unencrypted password over the network, so you should only allow access to private connectors via HTTPS.

The connector endpoints can then authenticate requests by checking the respective header field or query string parameter value.

To use a token in the request header, use the *Authentication* field to select *HTTP request header*, and enter a header name and header value.

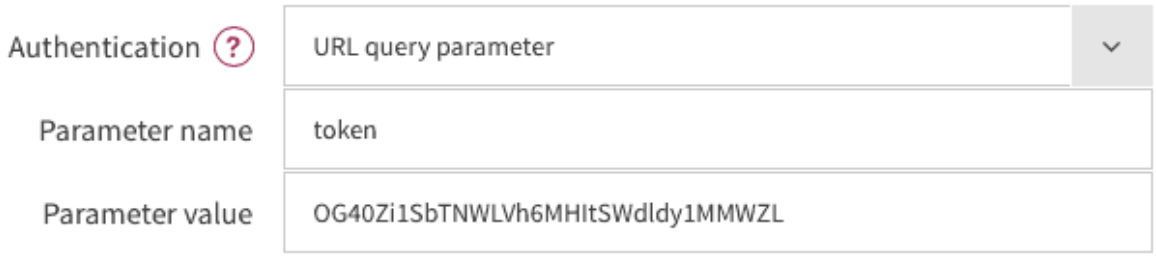


Configuring request header authentication

HTTP headers only allow a restricted subset of ASCII characters in header names, which typically only use letters and dashes, such as *Auth-Token*. Header values only support 'visible ASCII characters', so to allow arbitrary authentication tokens, use a Base64-encoded value. Configuring token authentication results in a request header like:

```
Auth-Token: OG40Zi1SbTNWLVh6MHItSWdldy1MMWZL
```

For testing, developers may find it more convenient to retrieve the authentication from the URL query string. To use this option, select *URL query parameter* and enter a parameter name and value:



Configuring URL query string parameter authentication

This results in HTTP requests with a URL query string, like this:

```
GET /?token=OG40Zi1SbTNWLVh6MHItSWdldy1MMWZL HTTP/1.1
Host: example.org
```

Warning: HTTP does not encrypt query string parameters, which typically appear in log files, so only use query string token authentication for testing a connector on a trusted network with the on-premise edition of Workflow Accelerator, and switch to a header field token for production use.

19.8 Connector examples

To help you develop your own connectors, Signavio has published several example connectors that show you what a connector implementation looks like. These examples use several different programming languages, including Java, Python, JavaScript, Scala and Go:

https://github.com/signavio/connector-examples

These examples have an open-source Apache License.

Chapter 20

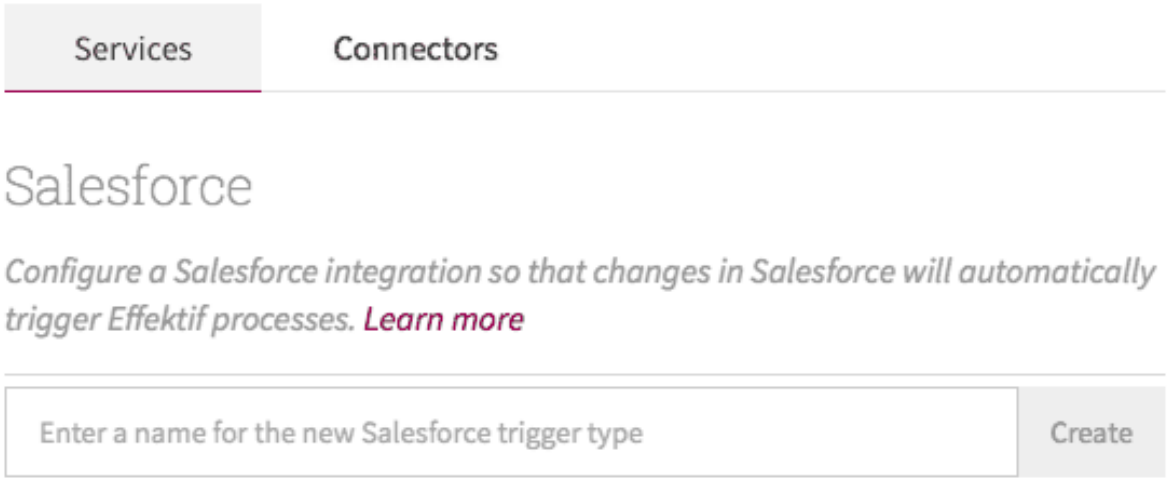
Salesforce Integration

Signavio Workflow Accelerator integrates with Salesforce workflows. You can configure this so that changes in Salesforce will automatically trigger Workflow Accelerator processes.

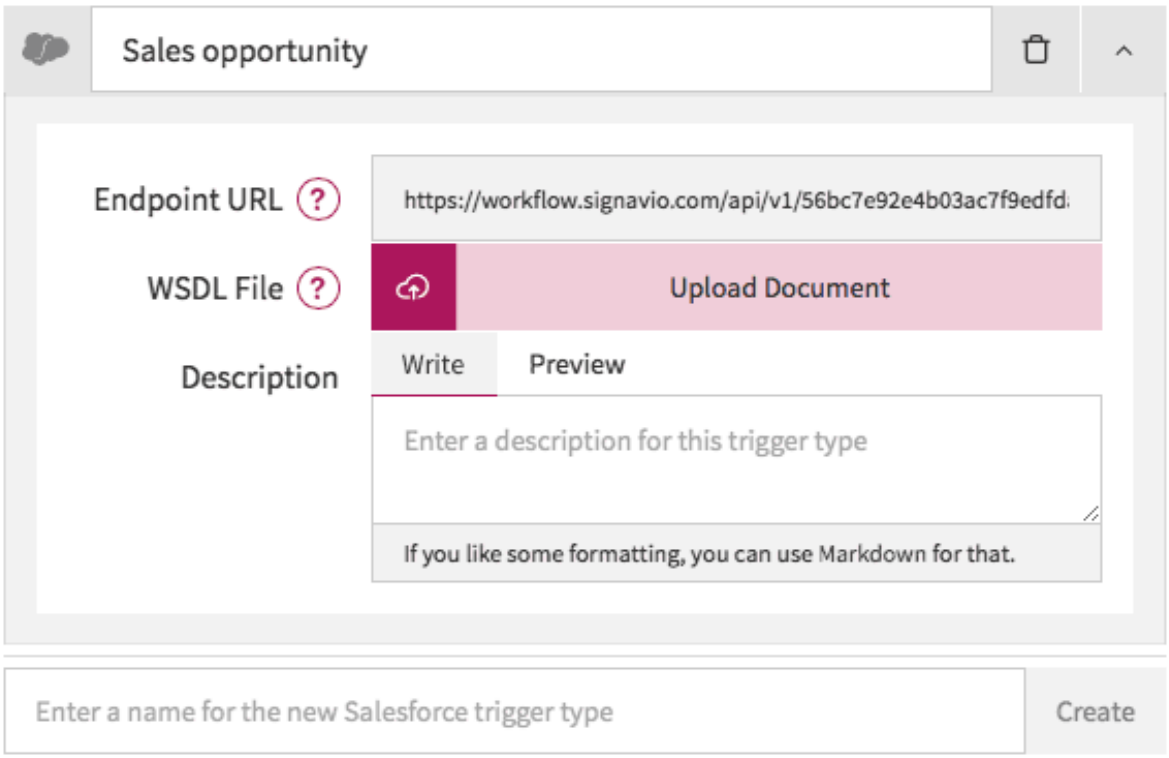
20.1 Workflow Accelerator configuration

Before you can set-up Salesforce integration, you need to create an endpoint in Workflow Accelerator that Salesforce can send information to. Set this up in Workflow Accelerator as follows.

In the top-right user menu, select *Services & Connectors*.



Under the Salesforce heading, enter a new trigger type name and select *Create*.



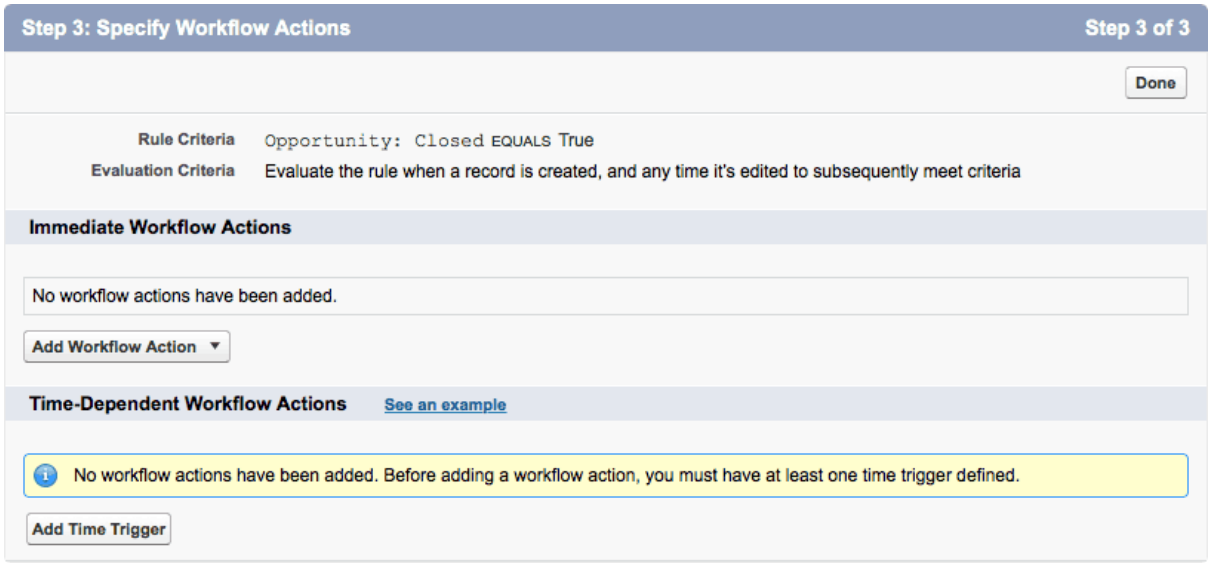
Copy the generated *Endpoint URL*, or leave this window open for later.

20.2 Salesforce configuration

Before you can enable Salesforce triggers, you need to configure Outbound messages - a kind of Salesforce workflow action. Set this up in Salesforce as follows.

Select the Setup menu item, and select *Build* → *Create* → *Workflow & Approvals* → *Workflow Rules*.

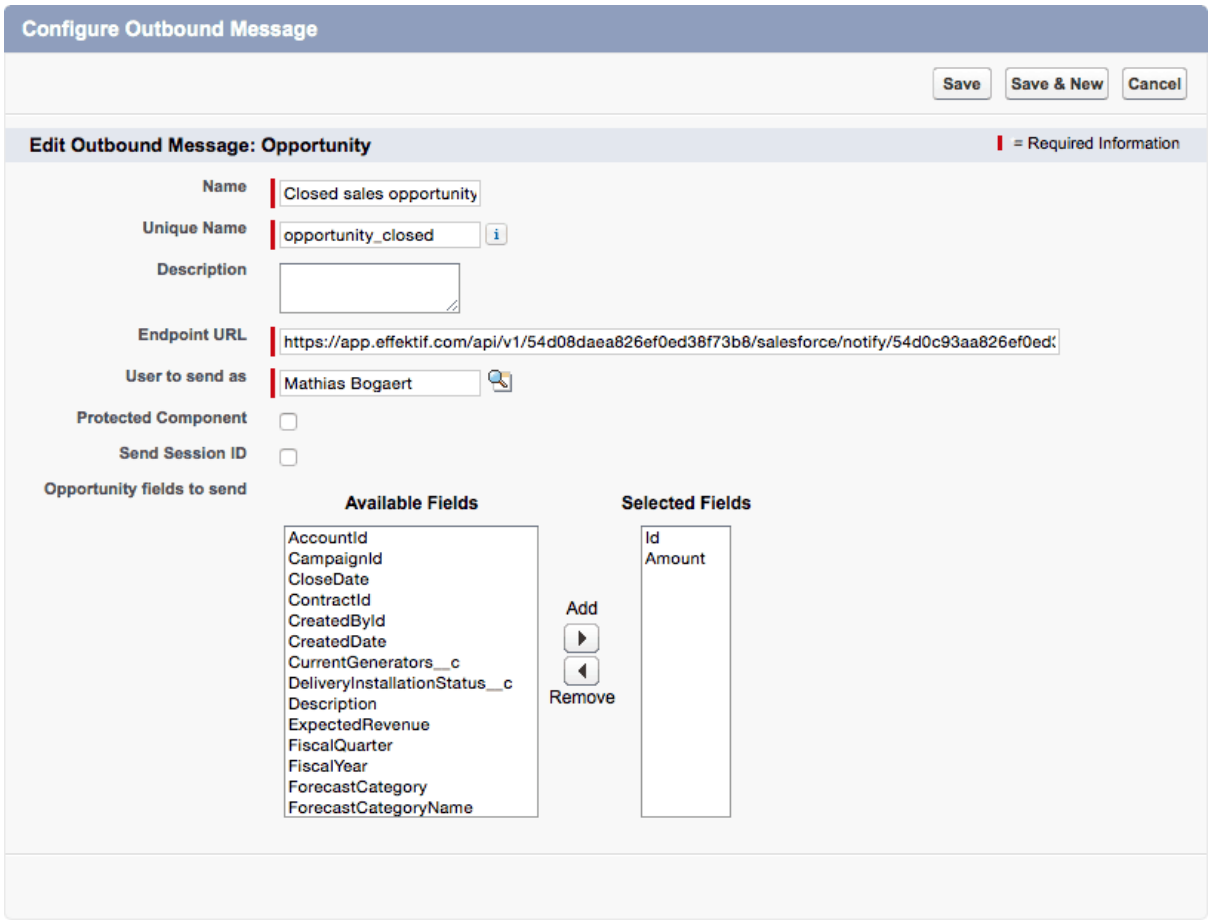
Create or edit a new or existing workflow rule, and in the *Workflow Actions* section, select *Edit*.



On the *Specify Workflow Actions* page, select *Add Workflow Action* and then *New Outbound Message*.

On the *Configure Outbound Message* page, configure the new *Outbound Message*, to define which object fields to send to Workflow Accelerator.

In the *Endpoint URL* field, paste the *Endpoint URL* from the Workflow Accelerator configuration you created earlier.



Select *Save*. Do not forget to activate the workflow rule as soon as you have finished the configuration. To do this, click the *Activate* button at the top of the *Workflow Rule Detail* page.

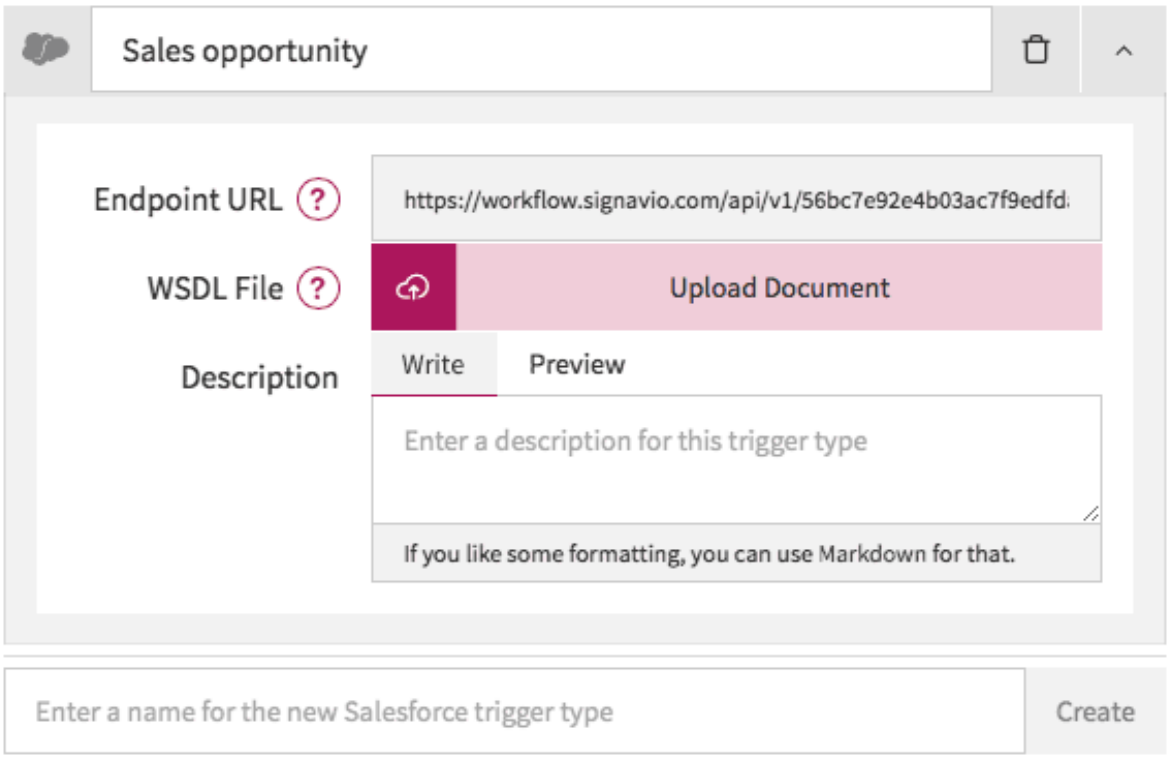
On the *Outbound message* page, next to *Endpoint WSDL*, select the *Click for WSDL* link, which opens a web service definition file.

Save the `workflowOutboundMessage.wsdl` file, which you will use to configure Workflow Accelerator, next.

20.3 Finish Workflow Accelerator configuration

For the final configuration step, use the web service definition file to configure the Salesforce service in Workflow Accelerator.

In the top-right user menu, select *Services & Connectors*, and then select the Salesforce trigger type you created earlier.



Select *Upload document*, and select the `workflowOutboundMessage.wsdl` file you saved earlier.

Now you can use the new *Salesforce trigger* (page 35) in Workflow Accelerator.

Chapter 21

Dictionary Integration

You can integrate the Dictionary from Signavio Process Manager to work with Signavio Workflow Accelerator. Doing so allows you to pull data from Dictionary entries and use them in your workflows.

Note: Your Signavio Workflow Accelerator organization needs to be connected to a Signavio Process Manager workspace before you can use this feature.

21.1 Activating the Dictionary integration

In Signavio Workflow Accelerator, select `Services & Connectors` from the top-right user menu.

Select Signavio Process Manager Integration, then select a user that also has a Signavio Process Manager account. Make sure this user is able to see all desired dictionary entries, as all requests to retrieve dictionary items will be done with this user. (Don't worry–you can always change this user later.)

In the dropdown, select which dictionary categories you want to use. Open one of these categories to see which fields are now available to use in Signavio Workflow Accelerator.

21.2 Using Dictionary categories with forms

In Signavio Workflow Accelerator, under the `Process` tab, open the process editor. Select a process element that requires configuration, such as a user task.

Under the `Configuration` tab, the categories you imported from Dictionary will be displayed as fields. Drag fields to your form to use them.

Now, when you execute your case, you will see a field where you can type and search for entries. Once you find the entry you want to use, simply click it to use it.

Hint: You can also use Dictionary entries in gateway conditions, script tasks, and emails the same as any other fields.

21.3 Additional info

Whenever you select an entry used in the task form during the case execution, Signavio Workflow Accelerator takes a snapshot of that entry. If the entry is changed later on, the snapshot in the case is NOT automatically updated. This is so that Signavio Workflow Accelerator can properly track past decisions that were based on dictionary entries. If a selected entry is set as the default value for a field, Signavio Workflow Accelerator will take a snapshot when the case is started.

Whenever you add or remove attributes to or from the dictionary category, you have to press the `Reload integration` button. The new attributes can then be used in the workflow editor after reloading the integration. They will become available as nested fields of the category fields.

Note:

- Attributes that have been removed from a dictionary category will still show up in old cases. You have to manually remove old dictionary attributes anywhere they are used in a workflow.
- Dictionary categories can be deactivated at any time. Once they have been deactivated, they can no longer be used in the workflow editor. You also have to manually remove deactivated categories anywhere they are used in a workflow. Old cases will still show data from deactivated categories.

21.4 Troubleshooting

When setting up the Dictionary integration, you may run into the following error message:

Could not set up the Dictionary integration for the following reason: The tenant ID is not configured for your Process Manager workspace. Please contact customer support.

To resolve this, you need to try to sync the configuration from Process Manager again. Open **Setup** > **Manage approval workflows** in Process Manager and click **Synchronize configuration now**. If that option is not available, you should contact the Signavio Support Team to set the tenant ID for you.

Chapter 22

Technical notes

Signavio Workflow Accelerator supports all popular browsers. At https://www.signavio.com/browser-compatibility you find a detailed description of the supported browsers. The site also automatically checks whether you browser supports the Signavio products.

Note: Since Microsoft stopped supporting Internet Explorer 9 and 10 for most platforms on January 12, 2016[44], it is getting increasingly difficult for Signavio to provide a state-of-the-art user experience while supporting these browsers. As we want to continue innovating and provide the best features and user experience possible, we will stop supporting Microsoft Internet Explorer 9 and 10 on December 31, 2017. If you use Internet Explorer 9 or 10, we recommend switching to one of the following browsers as soon as possible:

- Google Chrome[45]
- Microsoft Edge[46]
- Mozilla Firefox[47]
- Internet Explorer 11 (only if none of the above is viable)[48]

[44] https://support.microsoft.com/en-us/help/17454/lifecycle-faq-internet-explorer
[45] https://www.google.com/chrome/browser/desktop/index.html
[46] https://www.microsoft.com/en-us/windows/microsoft-edge
[47] https://www.mozilla.org/en/firefox/new/
[48] https://www.microsoft.com/en-us/download/internet-explorer-11-for-windows-7-details.aspx

Chapter 23

Tutorials

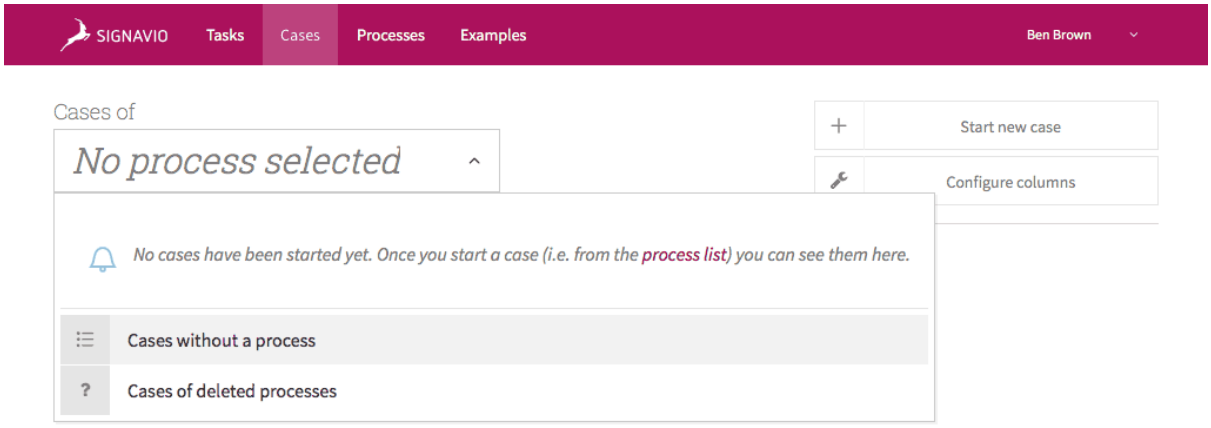
Use these tutorials to learn how to get started with Signavio Workflow Accelerator.

23.1 Using an ad hoc case for a document approval

This tutorial introduces the simplest way to get started with Workflow Accelerator. You can start without first defining a process by *Starting an ad-hoc case* (page 13).

People often use workflows for document approvals, so this tutorial uses the example of approving a report, called *June report*.

In the main menu, select *Cases* and then select *Cases without a process* from the bottom of the drop-down list. This shows the *Cases* (page 14) view, which you can use to create and view *ad hoc cases* (page 13).



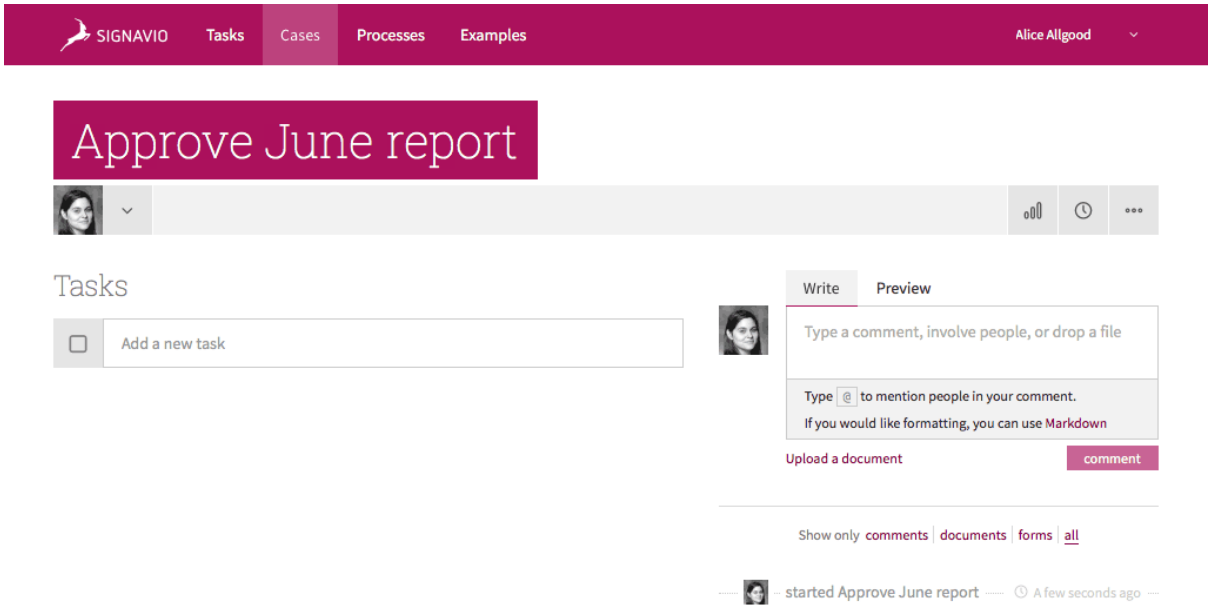
The Cases *view, where you can start a new case*

Note: *Ad hoc* cases do not use a pre-defined process model. You can also *create an approval process model* (page 172).

Select *Start new case* to start creating a new ad hoc case. This opens the case name prompt.

Entering a case name

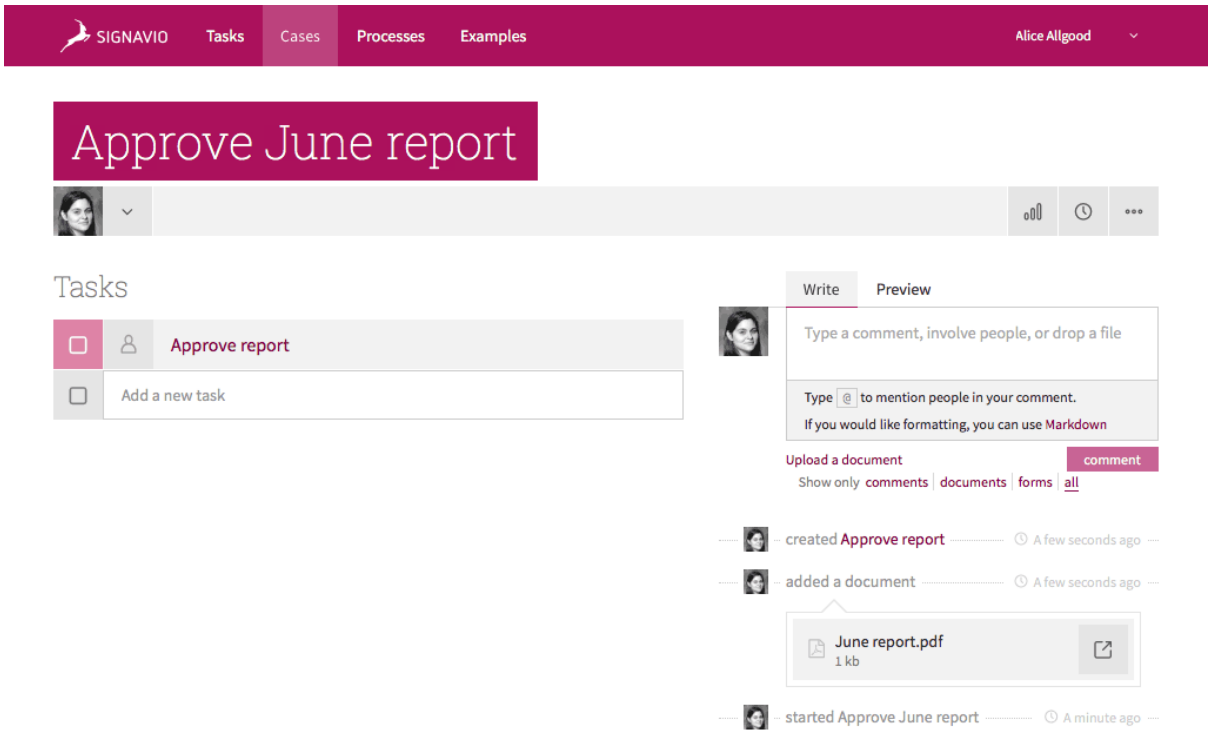
Enter the name **June report** to create the new case. The case details view shows the initial case, with an empty task list on the left.



A new ad hoc case

Now add a document: on the right, select *Upload a document* and select the report to review, **June report.pdf**. On the left, in the *Add a new task* text input, enter the task name **Approve report** to create a task.

The history panel now shows the corresponding events, labelled *added a document* and *created Approve report*.

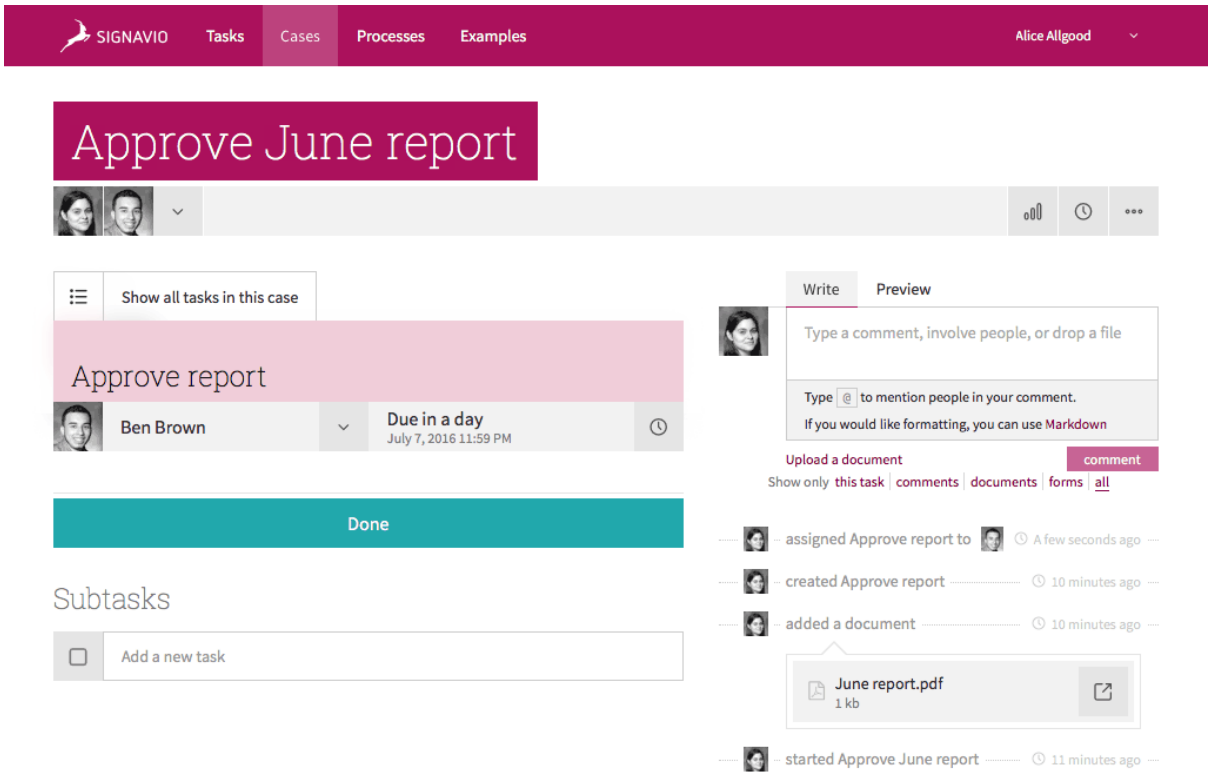


Adding a document to the history panel and a task to the task list

Note: You can add as many tasks to the case as you like. Use separate tasks for work that different people will do, or work that will they will complete at different times.

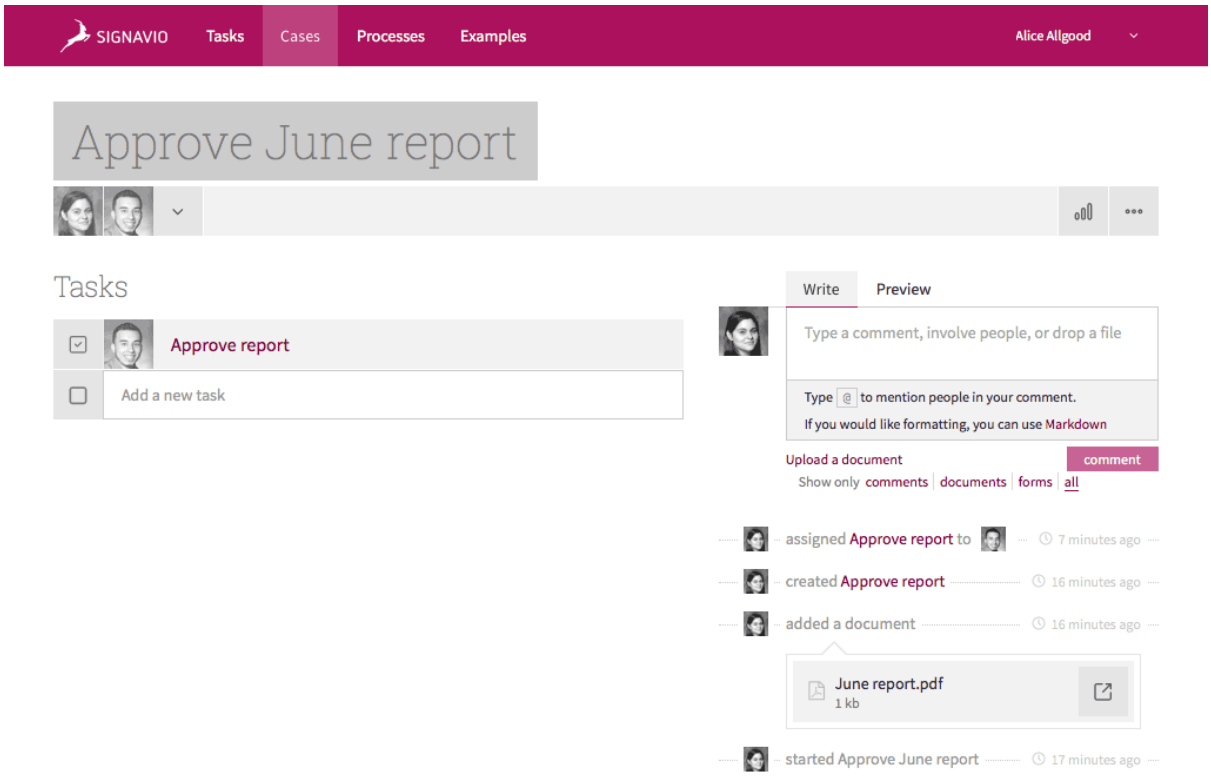
You can also use the text box above the comment panel to *add comments to the case* (page 19), to add information and collaborate with other people.

Select the *Approve report* task in the list to open the task details view. Under the task name, use the assignee selector to assign the task to someone, who will receive a *notification* (page 120). Use the date selector to choose a due date, which will result in reminders if the case's assignee does not complete it in time.



Details of a task within a case

Select the *Done* button to complete the task. As the case does not contain any other open tasks, this closes the case as well. You can recognize the case's closed status by the grey case name background, and from the most recent event in the event stream.



A closed case, after completing its last task

Note: You can use a similar case for any other kind of approval. Use comments to add any required information, and add approval tasks the same way. Next steps:

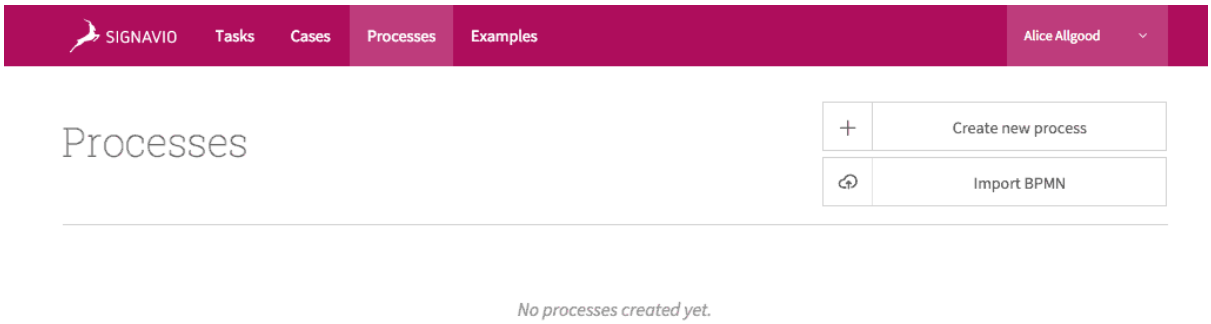
- use an ad hoc case for another kind of collaboration task
- use the *process builder* (page 28) to define a template for a repeatable process.

23.2 Your first document approval process

People often use workflows for document approval, a kind of management approval. This tutorial uses the example of a recurring process for approving some kind of report, which has two parts:

1. defining the process that forms a template of tasks for approving reports
2. running the process - starting a new case that groups the tasks for approving one particular report.

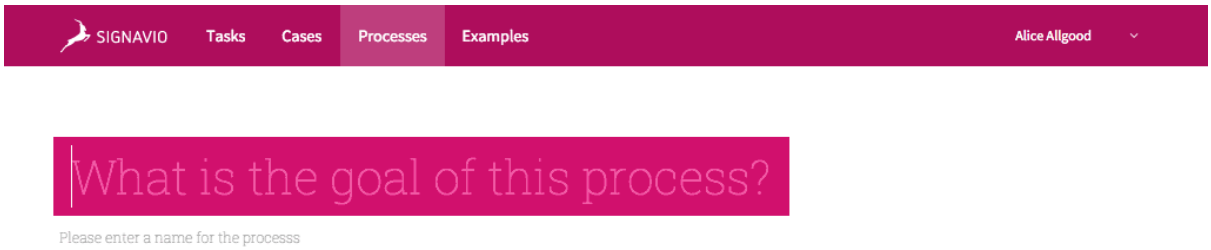
To get started, in the main menu, select *Processes*. This shows the *Processes* view, which you can use to create and view *Processes* (page 28).



The Processes *view, where you can create a new process*

Note: You can also *use an ad hoc case for a document approval* (page 168) without a pre-defined process.

Select *Create new process* to start creating a new process model. This opens the process name prompt.



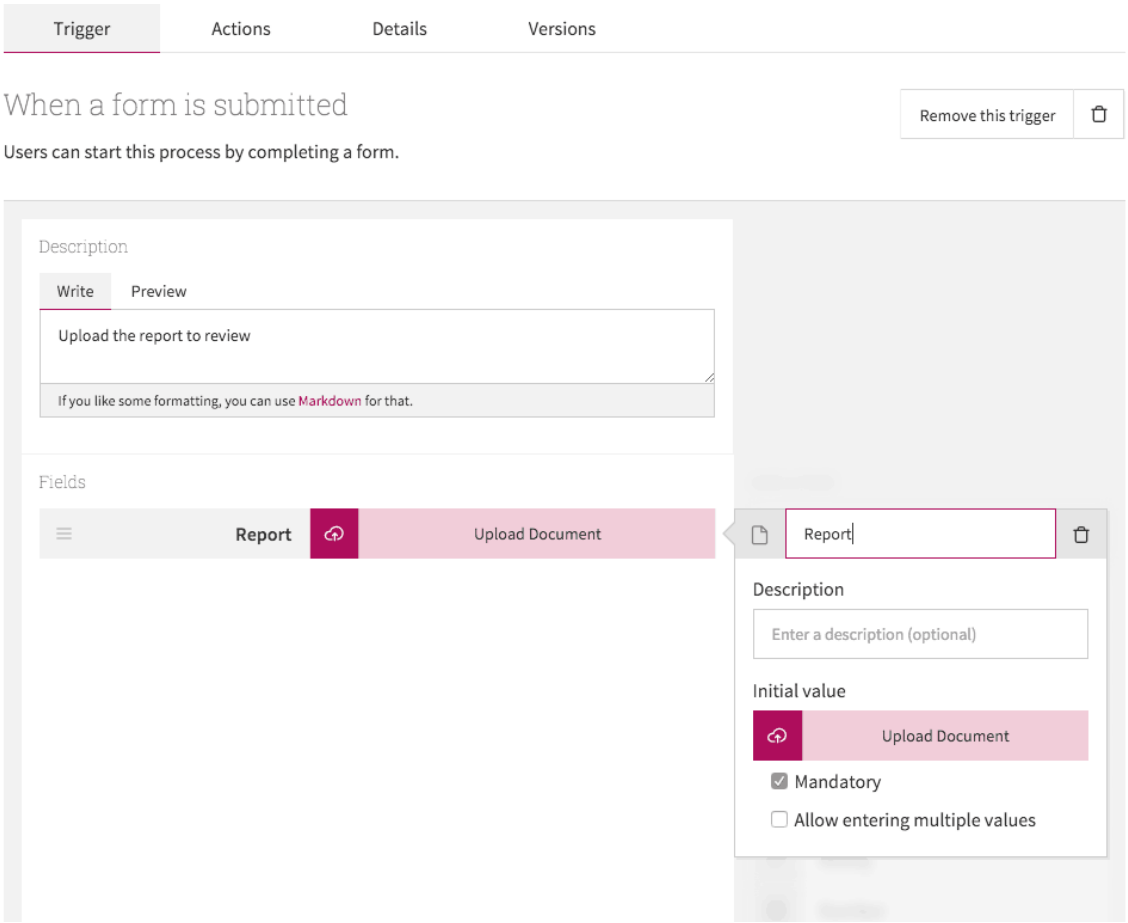
Entering a process name, to create a new process

Enter the name **Approve report**, which describes the process' goal. This creates the new process and opens the process builder's *Trigger* tab, which you use to define how the process starts.

On the *Trigger* tab, select *When a form is submitted* to add a form trigger, so you can start running the process by filling in a form. The document approval process requires a report to approve, which corresponds to a trigger form field called *Report* that you will use to upload a file.

In the form builder palette, select *File* to add the field to the form. Then select the field to open its configuration panel on the right, enter the field label **Report** and select the *Mandatory* option so the

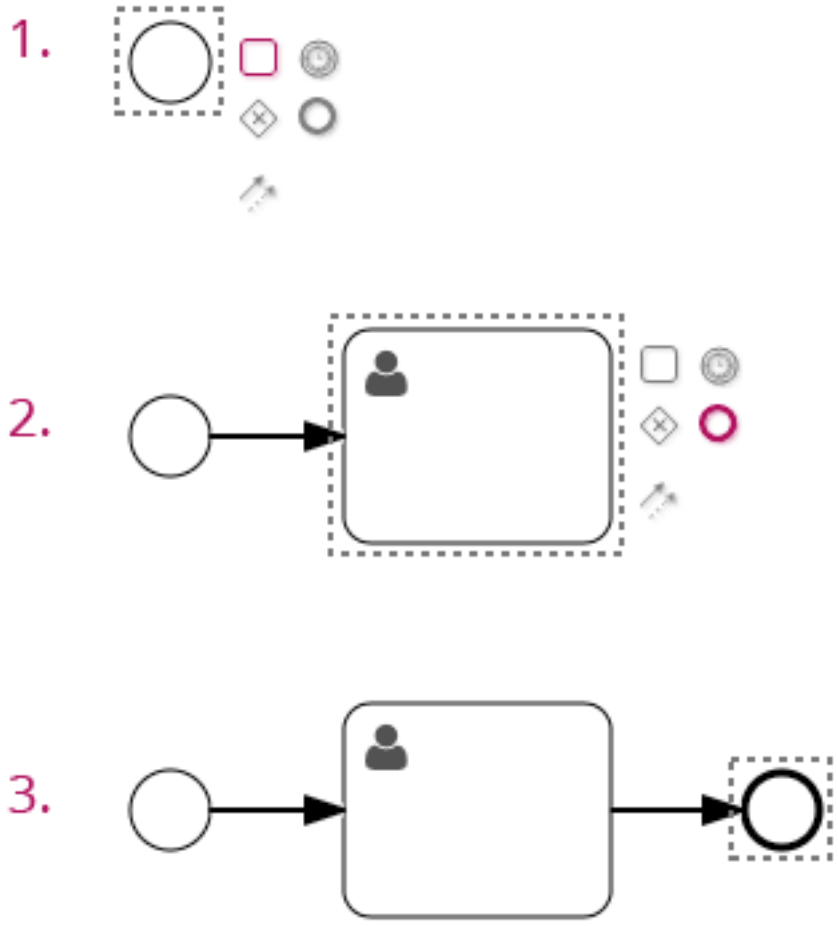
form requires a file upload.



The Trigger *tab, after defining a form trigger to start a case by uploading a file*

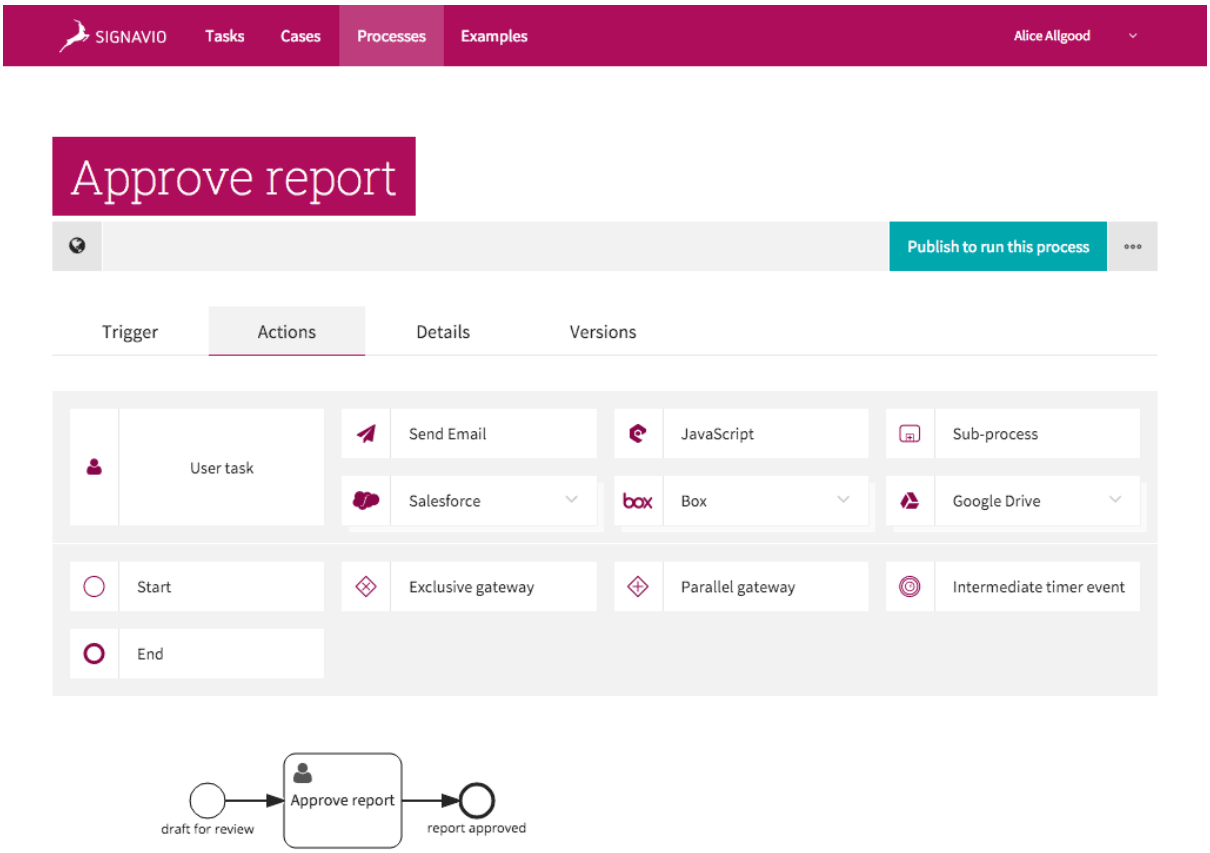
After choosing how the process starts, next define the actions that you will perform when running the process.

Select the *Actions* to load the graphical process editor. In the actions palette, select *Start* to add a start event to the diagram. Then, with the start event selected use the actions palette or the mini palette that appears when you select a diagram element to add a user task and end event.



Adding a start event (1), clicking the start event mini-palette's rectangle icon to add a user task (2), and clicking the user task mini-palette's circle icon to add an end event (3)

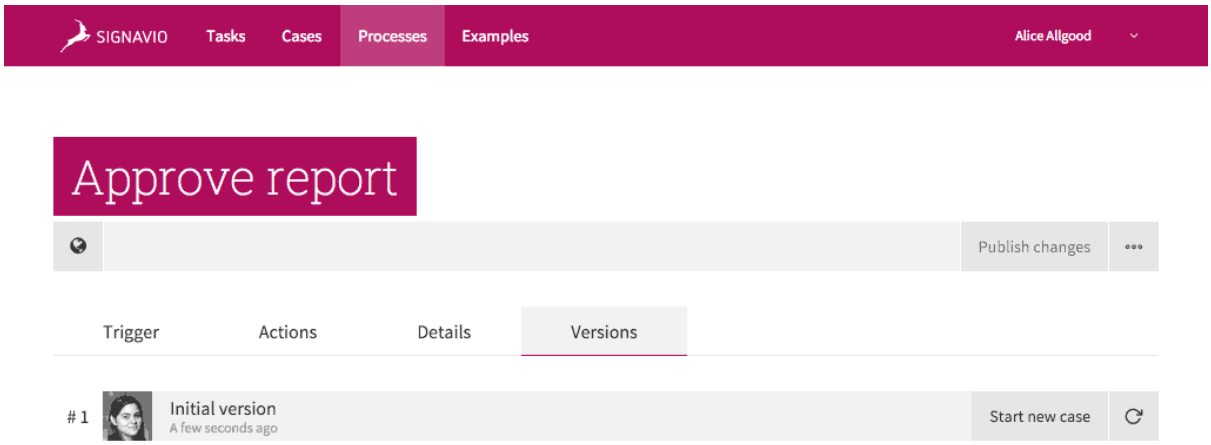
Next select the start event, user task and end event in turn, and use the configuration panel to set their names to *draft for review*, *Approve report* and *report approved*, respectively.



The Actions *tab, after adding a single* Approve report *action to a process*

This simple process model only contains a single task, to approve the report. Models don't have to contain start and end events, but their names help clarify the start and end statuses. Later, you can improve the workflow in various ways, but first you should run the process that you have defined so far, so you can see how it works.

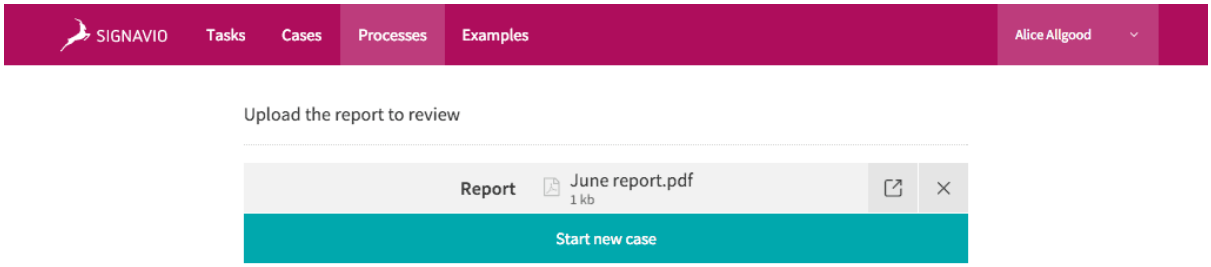
Select *Publish to run this process*. This creates a published version of the process, and shows the *Versions* tab, with this initial version.



The Versions *tab, after publishing the first version of a process*

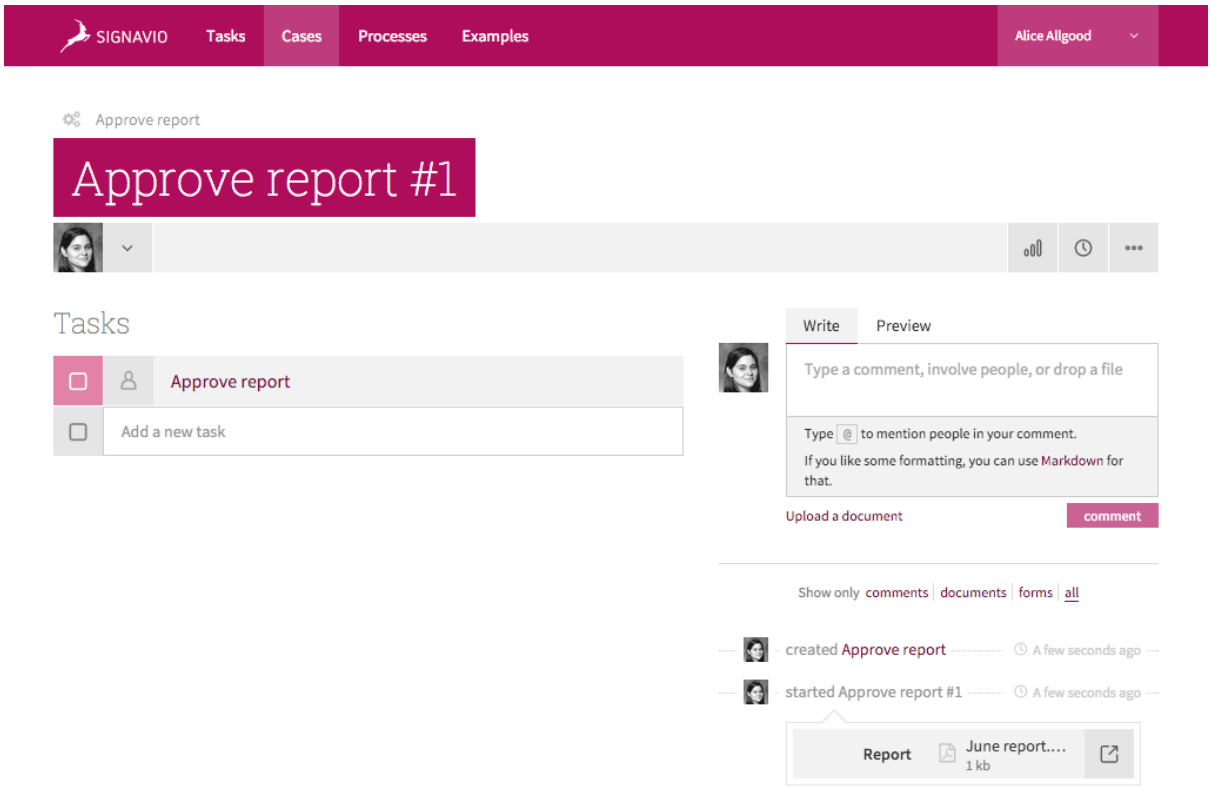
Now that you have published the process, you can use it as a template to create the first 'case' for approving a document.

Select *Start new case* to start a new case. This shows the trigger form you set-up earlier, which consists of a file upload field and a submit button. Select the file field, and choose a *June report.pdf* file to attach to the case.



Running the process - using the trigger form to start a new case

Select *Start new case* to finish starting the new case. This creates the case, and shows the case details view where you already see the process's *Approve report* task in the task list on the left. The first entry in the event stream, on the bottom-right, shows the the trigger form data, including the uploaded file, which you can select to open.



The case details view, after starting a new case

Now you have created an run your process for the first time, you can repeat the same steps to develop your process further: select *Processes*, select the process from the list, make changes to the process model in the process editor, publish a new version and then start a new case to try out the updated process.

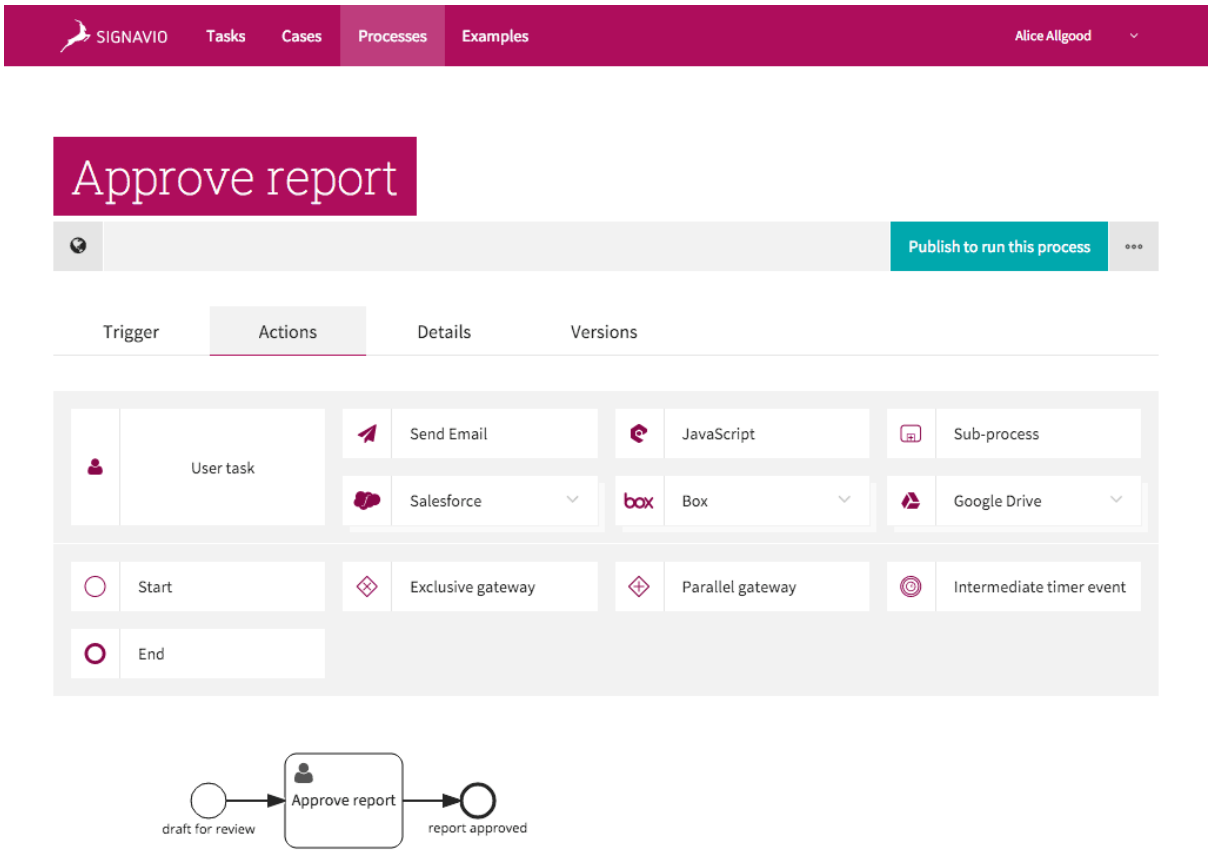
Note: After creating and running a simple approval process, you can enhance it in several ways. Next steps include the following.

- Adding an explicit approval decision using an exclusive gateway
- Adding a result notification using the send email action
- Using organization groups to define task candidates
- Using process roles to automatically assign tasks
- Using access control to restrict process actions

23.3 Adding a decision to an approval process

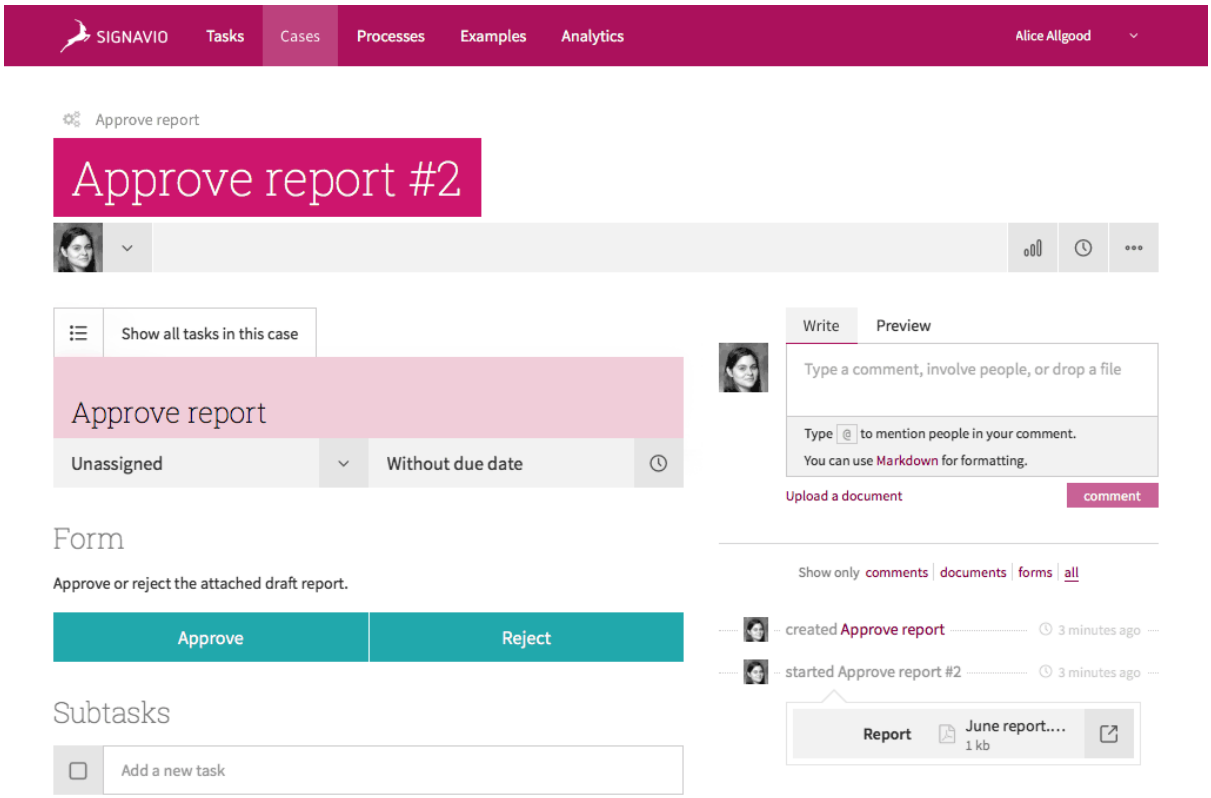
An approval process such as a document approval requires a clear decision, such as whether to *Approve* or *Reject* a document. This tutorial continues the document approval process example from the previous tutorial and shows you how to add a manual decision to a user task form.

To start, create a basic approval process with a single user task, as in the *first document process* (page 172) tutorial:



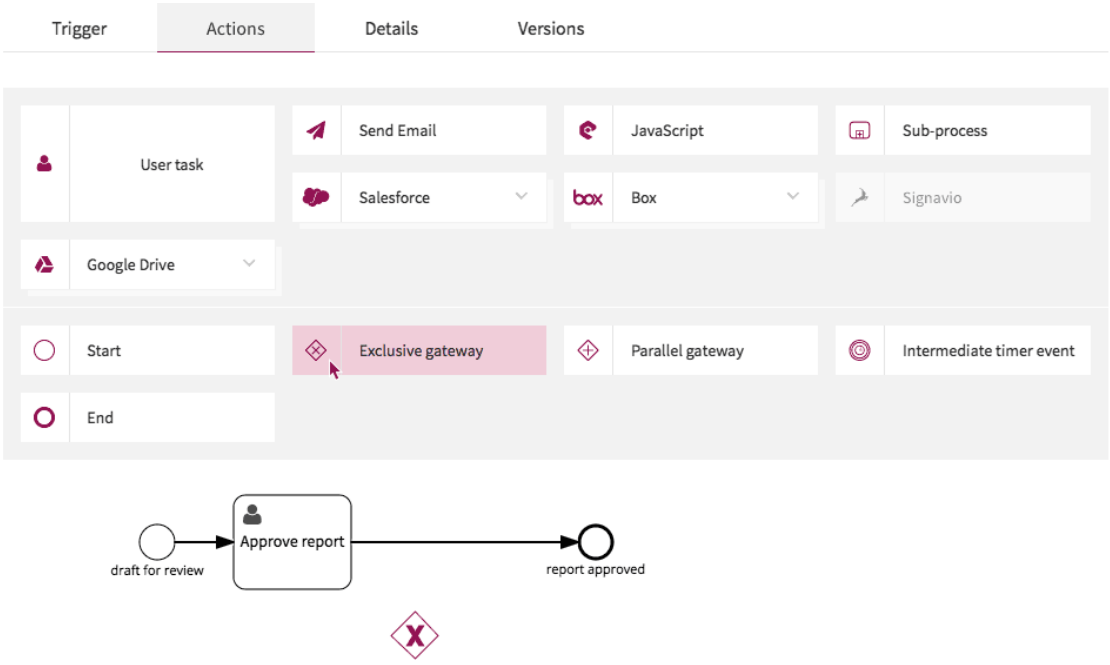
A basic approval process with a single task

This basic process already includes the task for making an approval decision, but it doesn't give any guidance for making the decision. You can improve this process so that the approval task's form has *Approve* and *Reject* buttons, like this:



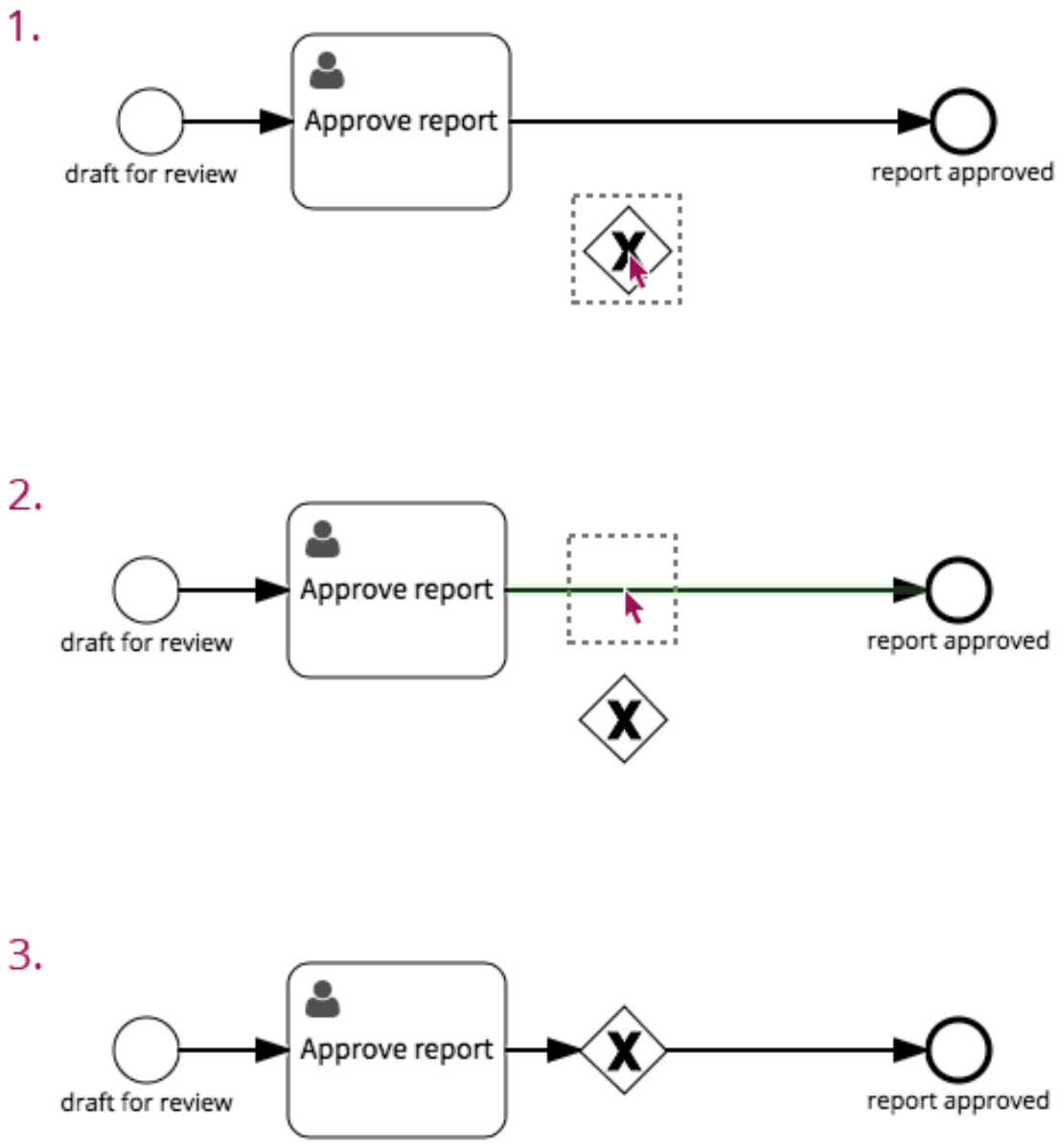
The result of adding a decision - a task form with Approve *and* Reject *buttons*

In the process model, an *Exclusive gateway* (page 97) after the user task will represent the decision. To add the gateway to the model, select the *Exclusive gateway* button in the tool palette. This adds the diamond shape with an X to the diagram.



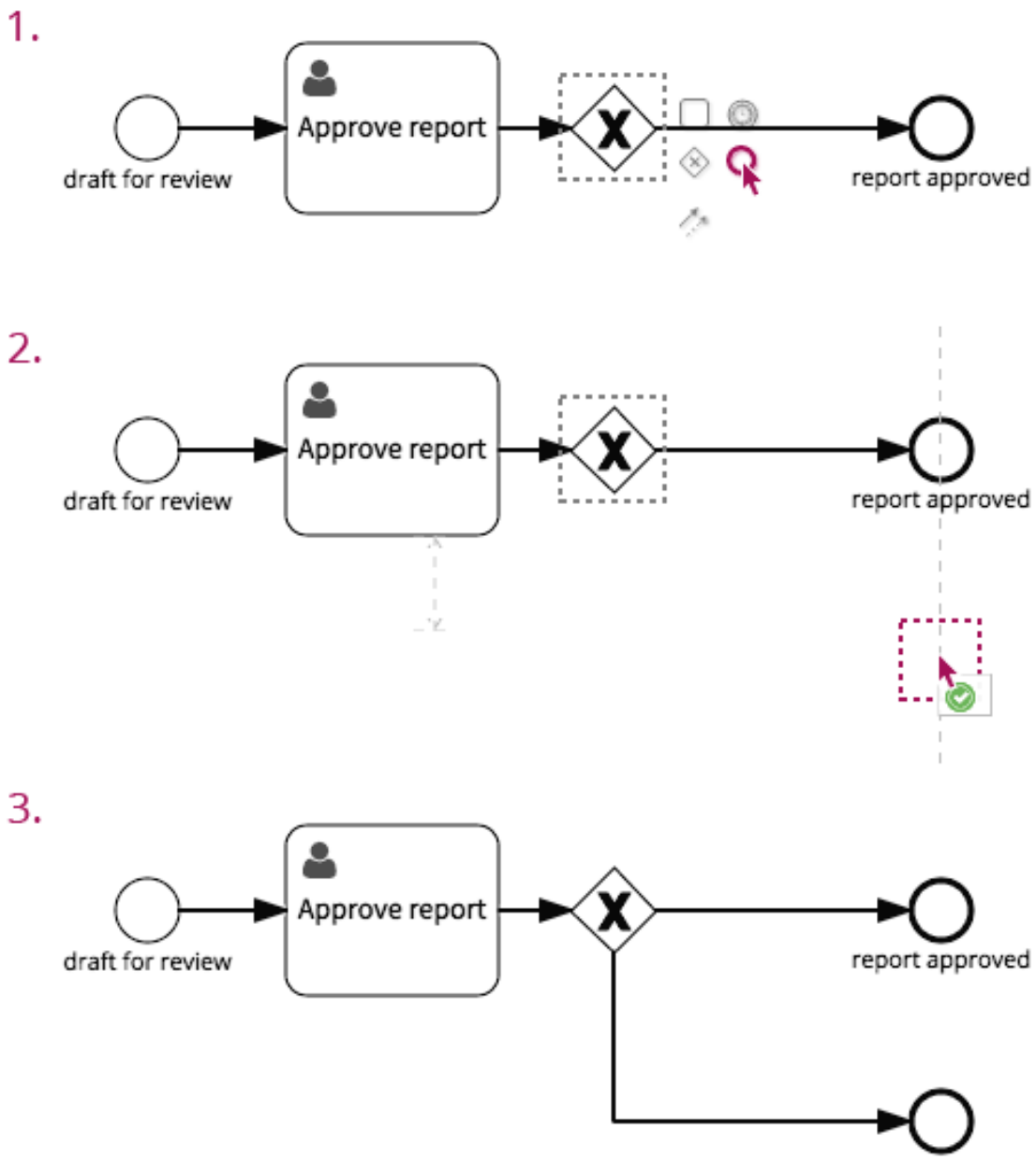
Adding an exclusive gateway to the process model

Next, drag the end event to the right, to make room for the gateway, and drag the gateway symbol onto the transition from the user task to the end event as shown:



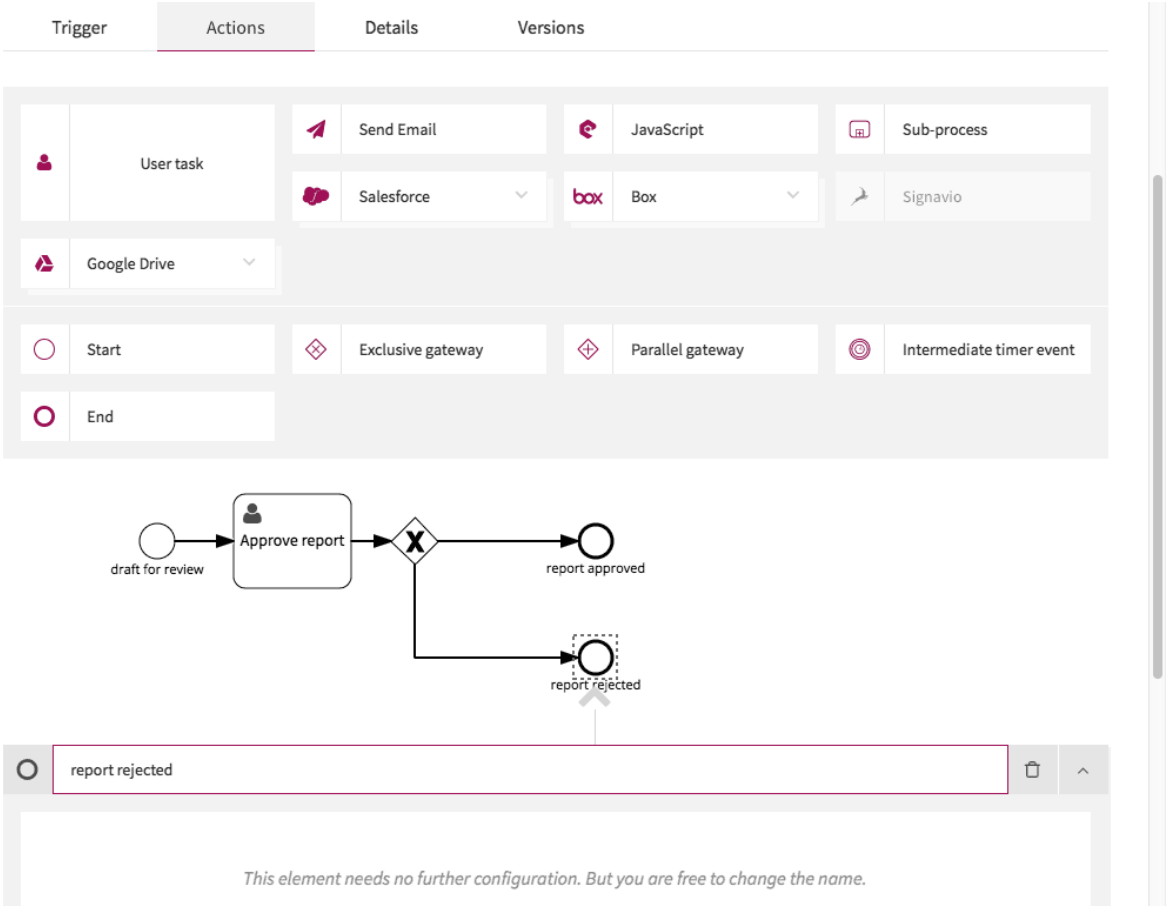
Moving the gateway to an existing transition

For the next step, add a new path to the process that represents the decision to reject the document. This means adding a second transition from the exclusive gateway to a new end event. To do this, select the exclusive gateway, and drag the end event (circle) icon to where you want to new end event, as shown:



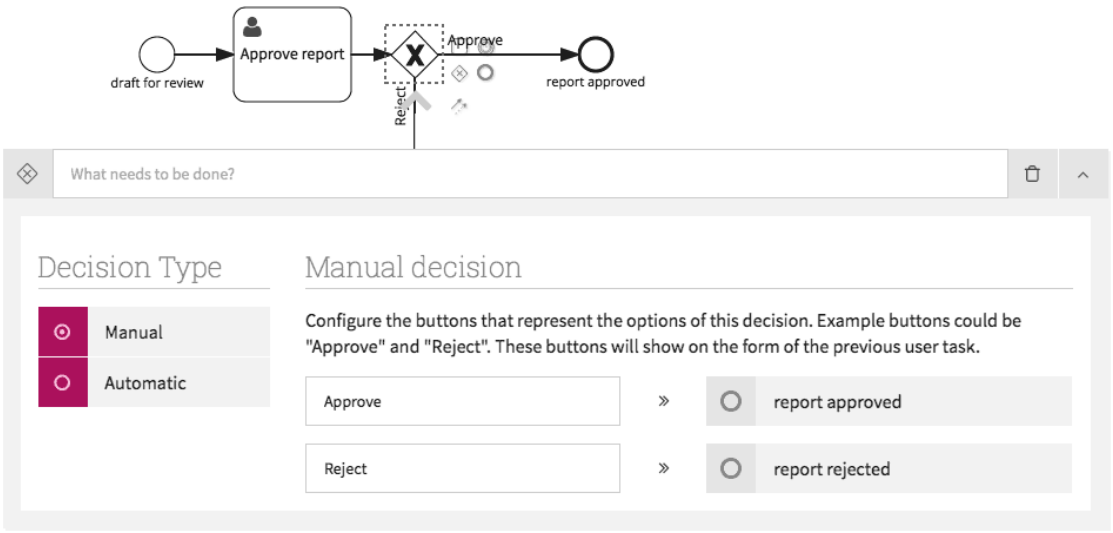
Adding a transition from the gateway to a new end event

Name the new end event to describe the alternate end status, to make the diagram easier to understand. Select the event and enter the name **report rejected**.



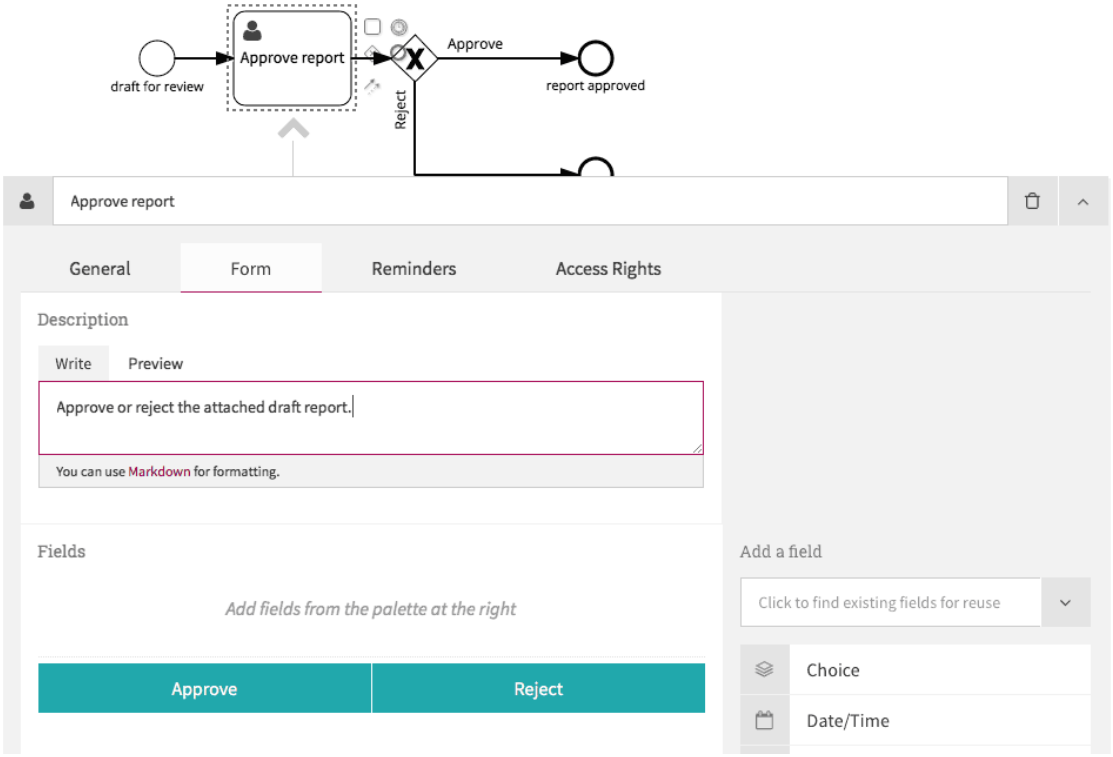
Naming the alternate end event with a different end status

Now you can configure the gateway with the decision. To use an exclusive gateway for a manual decision, it must have an incoming transition from a user task and more than one outgoing transition. Select the exclusive gateway to open its configuration pane, and enter the decision options **Approve** and **Reject**, using the end event names to get them the right way around.



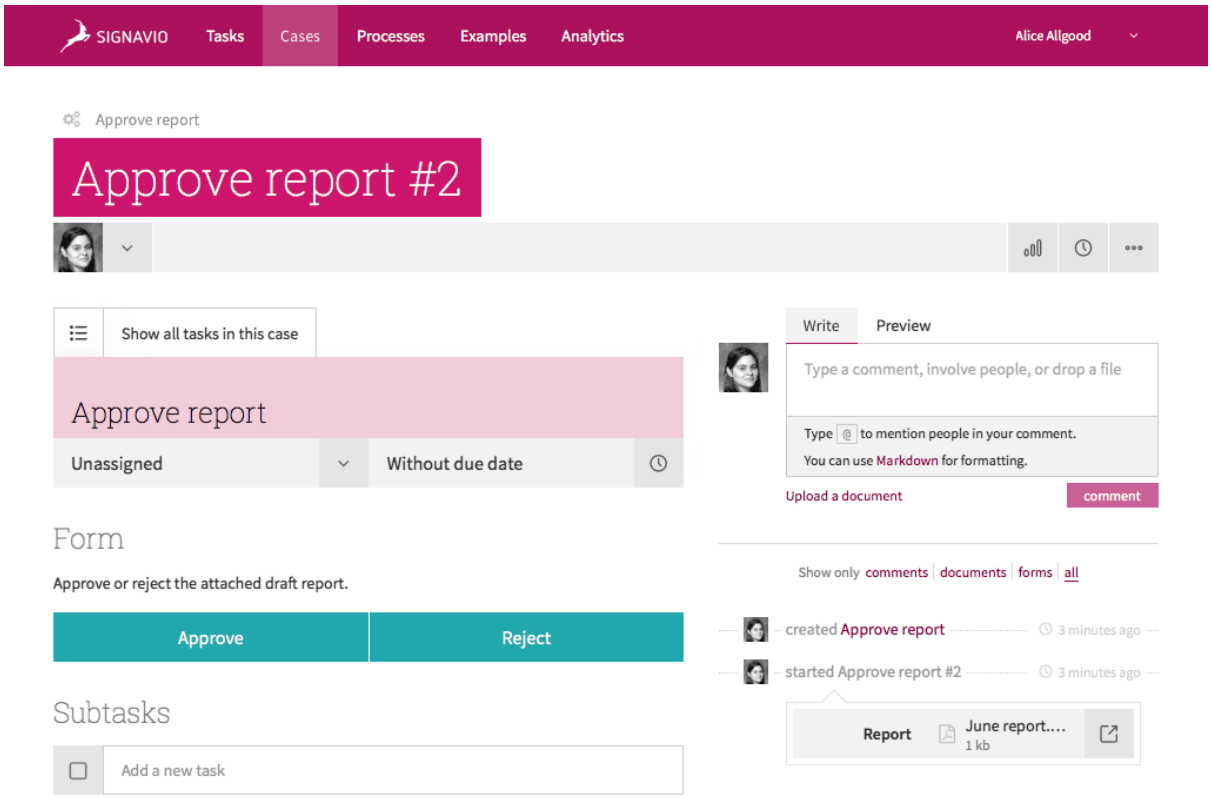
Configuring Approve *and* Reject *decisions on an exclusive gateway*

You can see the result of configuring the manual decision on the user task form. Select the user task, which opens its configuration pane's *Form* tab. At the bottom, underneath where any fields would appear, you now see the decision options as *Approve* and *Reject* buttons. In the form description field, enter instructions for making the decision: **Approve or reject the attached draft report**.



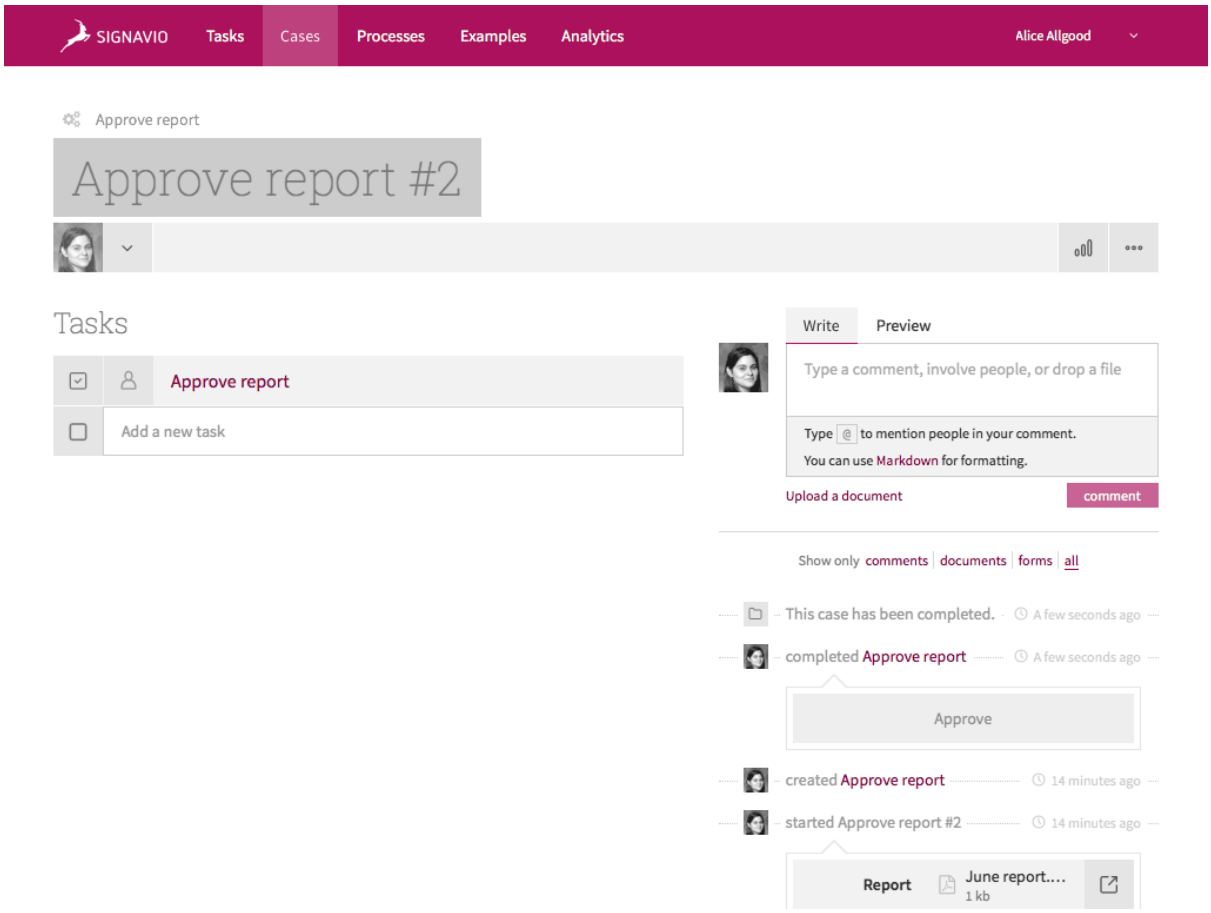
Adding a description to a task form that now shows Approve *and* Reject *buttons*

Now you can see the result of adding the decision to the process. Select the *Publish changes* button (top-right) to publish a new version of the process, then select *Start case* next to the latest version in the list. Start the case, completing the trigger form if you added one, and open the *Approve report* task. The task page shows the task form with the description you entered, and the decision buttons.



Running the process - the form for a manual decision with Approve *and* Reject *buttons*

Select *Approve* to record the decision and complete the user task. The case view history panel (right) now shows the *Approve* decision.



After completing the form, the history panel shows the Approve *decision*

Decisions like these don't only occur in document approval processes. In practice, many kinds of business processes use one or manual decisions that you can add in the same way.

23.4 More tutorials

If you need more examples on what Workflow Accelerator can do, Signavio's 10 Workflow Examples[49] page is a good place to start.

In addition, Signavio's Applied BPM Blog[50] includes Workflow Accelerator tutorials. The following tutorials introduce features based on concrete examples.

- Automatic workflow quality control tasks[51] - adding automatic review tasks
- Workflow web service integration[52] - updating an external database from a workflow
- Process milestones for Workflow execution status visibility[53] - using intermediate events to model process milestones
- DocuSign workflow integration[54] - adding electronic signatures to documents
- Multi-instance user tasks[55] - creating user tasks for members of a group

[49] https://www.signavio.com/post/10-workflow-examples/
[50] https://www.signavio.com/blog/applied-bpm/
[51] https://www.signavio.com/post/automatic-workflow-quality-control-tasks/
[52] https://www.signavio.com/post/workflow-web-service-integration/
[53] https://www.signavio.com/post/process-milestones/
[54] https://www.signavio.com/post/docusign-workflow-integration/
[55] https://www.signavio.com/post/multi-instance-user-tasks-workflow-accelerator/

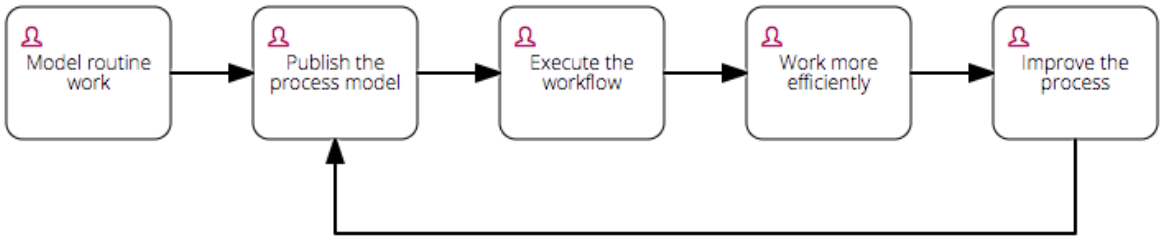
- Business days calculation[56] - using external public holiday data in JavaScript action
- Vacation handovers[57] - reassigning tasks and configuration task escalation
- Integrating a workflow with external web services[58] - fetching external data
- Integrating a spreadsheet with a workflow[59] - reading CSV data in a JavaScript action
- 'Four-eye' approvals[60] - adding permissions and parallel tasks to approval workflows
- Automatically triggering a workflow with form data[61] - reading from a trigger email
- Business rules execution from DMN[62] - executing decision models
- Custom activity types using sub-processes[63] - managing process complexity
- Decision gateway variables[64] - re-using decision results
- Form fields with multiple values[65] - entering lists
- Box.com integration[66] - uploading files to the cloud
- Google Cloud Print Integration[67] - printing files via the cloud
- Case name templates that identify orders[68] - organizing cases
- Role-based assignment[69] - configuring sticky task assignment

[56] https://www.signavio.com/post/business-days-calculation-workflow/
[57] https://www.signavio.com/post/smooth-vacation-handovers/
[58] https://www.signavio.com/post/workflow-web-services-integration/
[59] https://www.signavio.com/post/integrating-spreadsheet-workflow/
[60] https://www.signavio.com/post/automate-four-eye-approvals/
[61] https://www.signavio.com/post/automatically-trigger-workflow/
[62] https://www.signavio.com/post/business-rules-execution-dmn-model/
[63] https://www.signavio.com/post/custom-activity-types-sub-processes/
[64] https://www.signavio.com/post/decision-gateway-variables/
[65] https://www.signavio.com/post/form-fields-with-multiple-values/
[66] https://www.signavio.com/post/box-file-upload-integration/
[67] https://www.signavio.com/post/google-cloud-print-integration/
[68] https://www.signavio.com/post/case-name-templates-to-identify-orders/
[69] https://www.signavio.com/post/using-role-based-assignment-in-effektif/

Chapter 24

Introduction

Signavio Workflow Accelerator[70] is a web-based workflow modeling and execution platform. Although its heritage includes classical Business Process Management Systems (BPMS), Workflow Accelerator dramatically simplifies workflow automation.



You can use workflow automation for a variety of business processes - both for industry-specific processes, and for central functions such as human resources.

24.1 When to use Workflow Accelerator

You will find Workflow Accelerator useful for describing and collaborating on routine work. Use Workflow Accelerator for:

- coordinating tasks and handovers
- approvals
- routing documents
- fully-fledged business processes

24.2 Benefits

You'll get:

- control where you need it
- flexibility
- fewer delays (with automatic triggers, actions and timers)

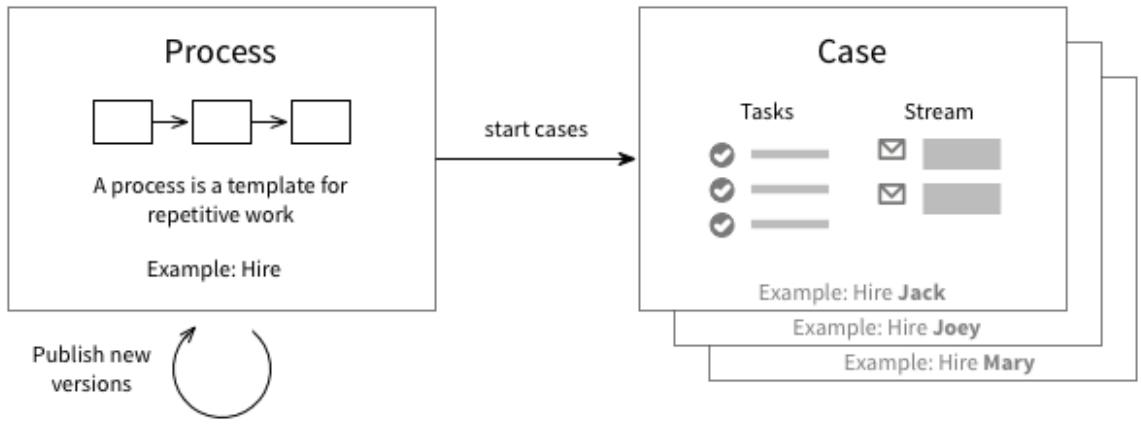
[70] http://www.signavio.com/products/workflow/

- no more miscommunications during handovers
- traceability - data on who did what
- clarity - visibility of who has to do what
- agility - because you can change Workflow Accelerator process models more easily than custom software

24.3 Do I need BPM knowledge to use Workflow Accelerator?

No. Workflow Accelerator is easy to use and aimed at anyone who needs to describe and collaborate on routine work, regardless of their familiarity or comfort with BPM. If you don't have BPM experience, you'll find it's still quite easy to get up to speed using Workflow Accelerator.

24.4 How it works



A process–specifically, a process *model*–is a template for repetitive work. Within an organization, processes are used to manage work and specify the tasks and actions that one must complete to reach a certain goal. For example: for every Hire employee[71] case, someone must complete the *Plan interview*, *Interview candidate* and *Send job offer* tasks.

After you publish a process, you can start many individual cases. Workflow Accelerator keeps track of which tasks and actions you have to perform for each case. Cases bring people together with relevant information that provides context for completing the tasks. You can also use cases as collaboration spaces for people to discuss and create new tasks for individual cases on the fly.

24.5 Examples

Browse the Workflow examples[72] to see different ways to apply workflow automation. You can use workflow automation for a variety of business processes - both for industry-specific processes, and for central functions such as human resources.

In the application itself, you can select *Examples* from the drop-down menu (top-right). On the *Examples* page, select *Copy to your organization* to create a copy of the example that you can edit to see how it works, and adapt to your own business process.

[71] https://www.signavio.com/workflow-examples/hire-employee/
[72] https://www.signavio.com/workflow-examples/

24.6 Acknowledgements

Signavio uses open source software. We thank everyone involved in the open source community. Please download the `list of open source components that Workflow Accelerator uses` (PDF).