



Processes and automation

ELO Flows



Table of contents

ELO Flows - Getting started	3
What is ELO Flows?	3
Quick start	5
Installation & operation	10
Overview	10
Architecture	11
Installation	13
Troubleshooting	15
Use	22
Flow administration	22
Automations	31
User actions	41
Data queries	55
Credentials	68
Webhooks and token authentication	71
Icons	74
Transformations	77
JSONata editor	81
Advanced flow	83
Synchronous flows	87
Components	94
What are components in the context of flows?	94
General information	96
ELO objects and metadata	100
ELO search	115
ELO feed	116
Time	117
ELO workflow	119
SMTP Mail	122
FTP	123

ELO Flows - Getting started

What is ELO Flows?



ELO Flows is an ELO module for easily implementing automation and integration tasks.

ELO Flows enables you to automate processes, define simple user actions, and integrate third-party systems without the need for any scripting. This enables you to realize processes based on low-code or no-code methods.

When should I use ELO Flows?

- You want to provide input fields in the ELO clients. These forms can be configured as a tile or as a button on the ribbon.
- You want to create automation tasks across services.
- You want to connect custom developments with ELO. It is possible to develop custom components.
- You'd like to integrate ELO in an existing system or integrate other applications in your existing environment, for example ERP systems, e-mail servers, file shares, etc.

Why should I use ELO Flows?

- With ELO Flows, you can combine functions from different components.
- When configuring flows, you can perform advanced definitions. A declarative transformation language (JSONata) with a content enrichment tool is available for this.
- With a worker model, you can scale job processing.
- You can combine existing components and develop your own custom components to meet individual requirements.
- ELO Flows is part of the ELO license.

What is the difference between ELO Flows and other ELO components?

ELOas

ELO Flows can be understood as the gradual successor to ELOas in the long term. ELOas will continue to be supported and maintained, but it is no longer being developed.

ELO Flows makes it easier to create automation tasks. Ideally, no custom scripting is required at all.

ELO workflows

In conjunction with ELO workflows, ELO Flows makes it easier to map business processes.

Technical logic can be implemented with flows, while business logic is handled via ELO workflows.

Linking these two modules creates a comprehensive process tool.

ELO Business Logic Provider (BLP)

ELO Flows will exist alongside ELO BLP.

ELO BLP will remain the method of choice for complex integration scenarios.

ELO Business Solutions

The ELO Business Solutions are applications based on the ELO Suite.

ELO Flows will provide the technical platform for future ELO Business Solutions, replacing parts of the Common framework.

Quick start

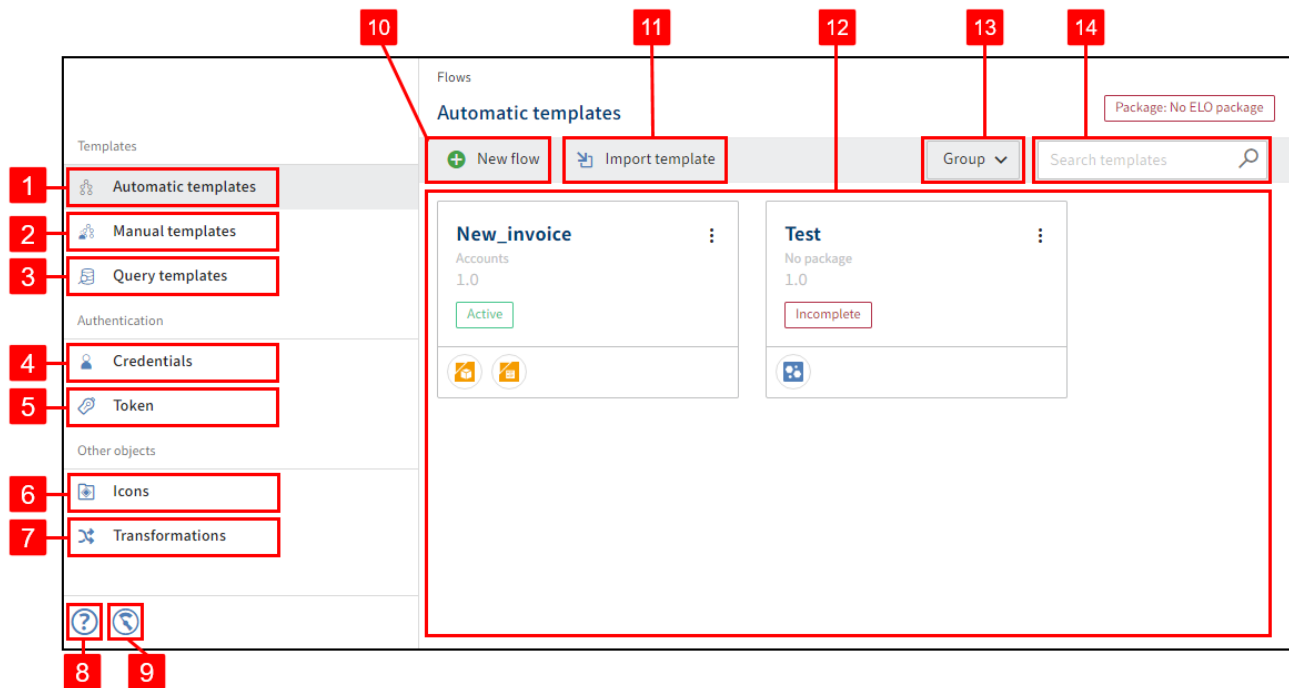
Welcome to ELO Flows. We've prepared a quick start guide for you to help you get started!

Deployment

ELO Flows is provided with the installation of the ELO Server Setup. If it isn't already installed, you can start the ELO Server Setup to do so.

Flow administration

You can access the flow administration area via the ELO Administration Console. ELO Flows is integrated in the administrative environment of [ELO packages](#). You can define and edit the flow logic in the *Flows* area.



You have the following options:

1 Create, edit, enable, and disable automations

You can find more information under Automations.

2 Create, edit, enable, and disable user actions

You can find more information under User actions.

3 Create, edit, enable, and disable data queries

4 Configure credentials for components

You can find more information under Credentials.

5 Create token: You can safeguard trigger events that are triggered by third-party system calls by creating a token with an assigned user context.

You can find more information under Webhooks and token authentication.

6 See and edit icons

You can find more information under Icons.

7 Create transformations

You can find more information under Transformations.

8 Help: Opens the help page in a new tab.

9 Monitoring: Opens the *Status Report* page. There, you will find information about the status of the executed flow.

You can find more information under Monitoring.

10 Create new flow: Create new user actions, automations, or data queries.

11 Import: Import exported flows as a JSON file.

12 Viewer pane: Here you can see the flows currently available.

13 Group existing flows

14 Browse existing flows

Flow types

You can choose between three flow types:

- Automations
- User actions
- Data queries

The difference is in the initial trigger.

Automations start with ELO events, among other things. If you install a component offering triggers, use them here. In addition to the ELOix events, the action of receiving an e-mail is conceivable with an e-mail component. You could respond to this and run a flow.

User actions start with a manual trigger. These allow users to make entries within an ELO client. This data is transferred to the flow and can be processed there.

Data queries start with a webhook or with ELO events.

Triggers

In ELO Flows, flows are started by triggers. There are different types of triggers.

- Automatic triggers: Among other things, these listen to ELOix events (provided by the ELOix event connector).
- Multi-instance trigger: These triggers enable flow instances to be processed in stacks. One flow instance is started for each object returned by the trigger.
- Manual triggers: These provide a button to manually start the flow in the clients. Behind this button is a definition for the activity UI, which the activities of an entry in the ELO client for ELO Flows map.
- Webhook: A webhook trigger creates a REST endpoint that can be called by external systems.

Services

Services are functions that are called in the flow steps.

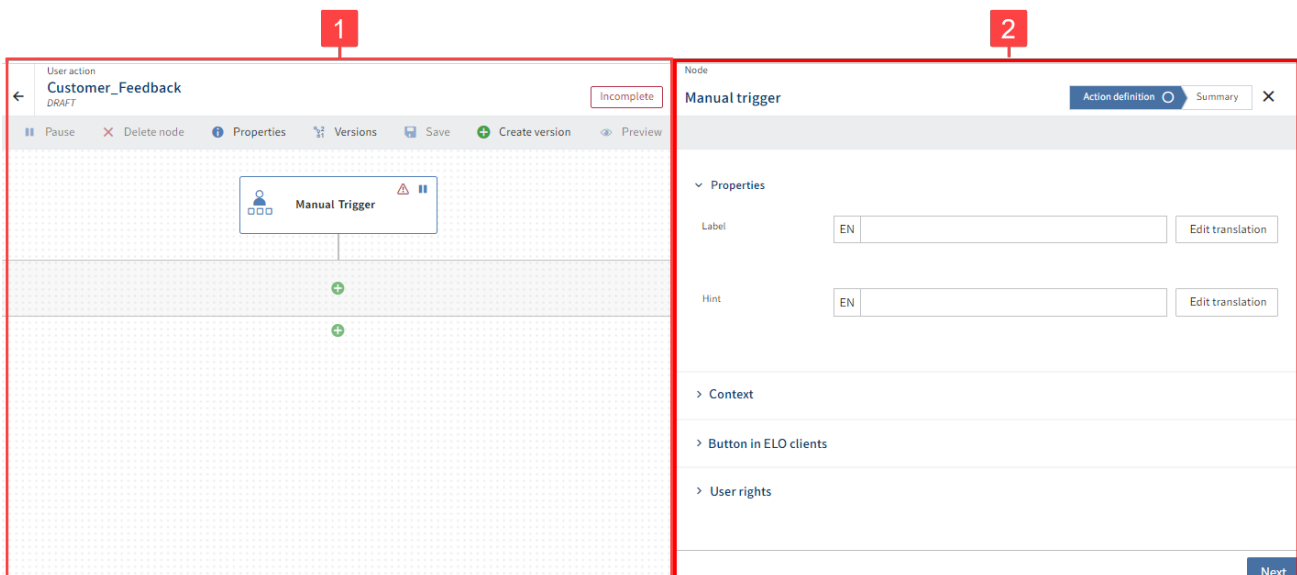
Activities

Activities are user actions that are opened with a button in the ELO clients. The button definition is configured for a user action. This allows you to quickly map common use cases, such as a form activity or a document upload activity.

Refer to *User actions* under Add activities for more information.

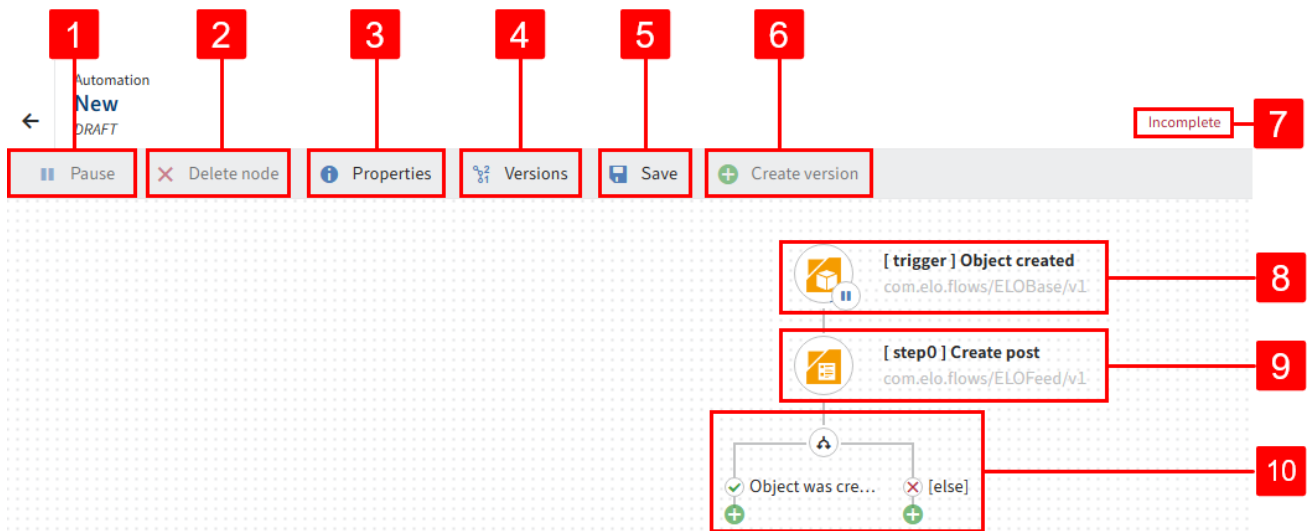
Flow designer

With the flow designer, you can edit existing or new flows.



1 Flow designer: Design custom flows and their properties by adding new nodes, deleting nodes, or saving, versioning, activating, or pausing the flow.

2 Configuration area: Select the component you want to integrate into the flow and configure the specific properties of the flow's nodes.



The flow designer contains the following elements:

1 Pause/Continue: Set a flow to *Active* or *Deactivated*. Triggers are not called for deactivated flows. For user actions, the respective buttons are hidden in the client.

2 Delete node: In the flow designer, triggers, services, and branches are shown as nodes. When configuring a flow, you can select existing nodes and delete them from the flow using *Delete node*.

3 Properties: You can change the name of the flow, as well as export and delete the flow here.

4 Versions: You can select the versions to delete them, edit them in the flow designer, or activate or deactivate them. When you edit a flow, it will initially be labeled with *DRAFT*. Created versions can be found under the name you assigned them. You will find more information on the versioning of flows under Use > Advanced flow.

5 Save: Save regularly so no changes are lost. If the flow is activated, saving results in hot deployment in the ELO client.

6 Create version: Create a new backup of the current version. Enter a version name. You can also add a comment to the version.

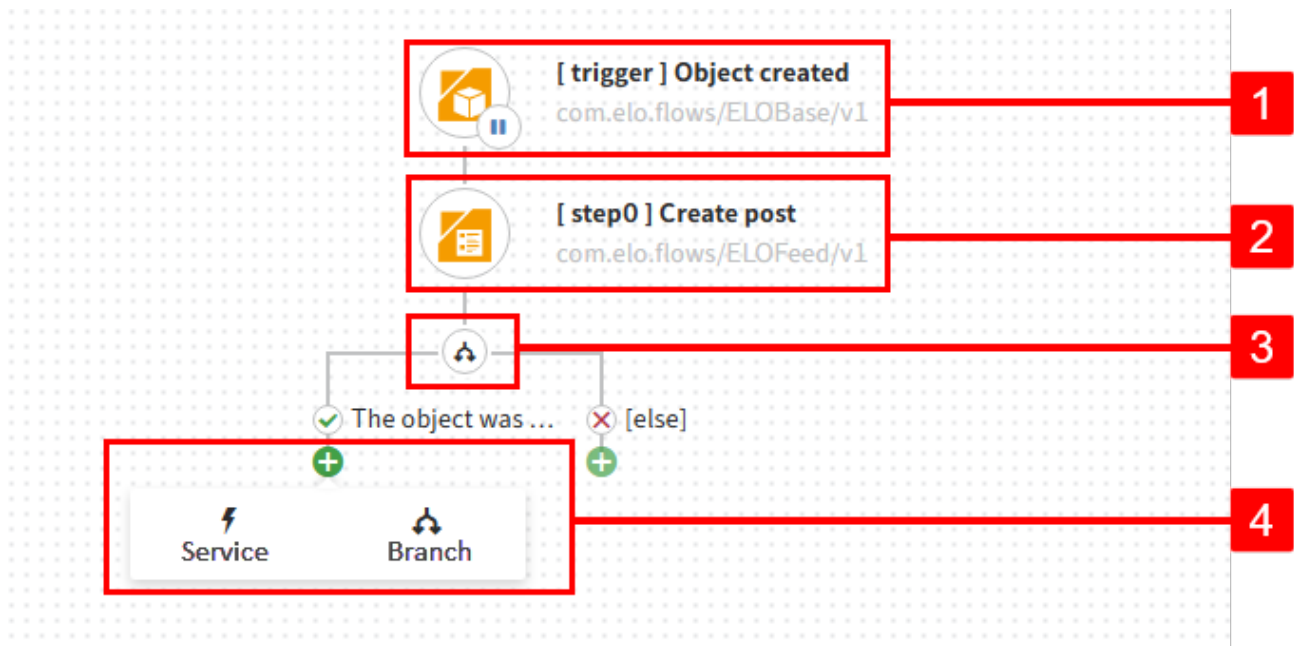
7 Flow status: *Active*, *Deactivated*, or *Incomplete*

8 Trigger node

9 Service node: The additional *[Step]* signifies the order of service nodes you have assigned to the flow.

10 Branch: You can also create a branch with a true-false condition before each step. You can nest this branch in other branches or continue with services.

Structure of a flow



A flow consists of the following elements:

- 1 Trigger: Starts the flow
- 2 A series of services: Steps in the flow
- 3 Branch node: True-false condition that determines how the flow proceeds
- 4 Add node: Adds a service or a branch

Components

Components are connecting modules or connectors to systems. You can use standard components that are provided, such as the *ELO metadata and objects* component, or create your own components that interact with the systems you want. You will find more information on the standard ELO Flows components in the *Components* section.

Installation & operation

Overview

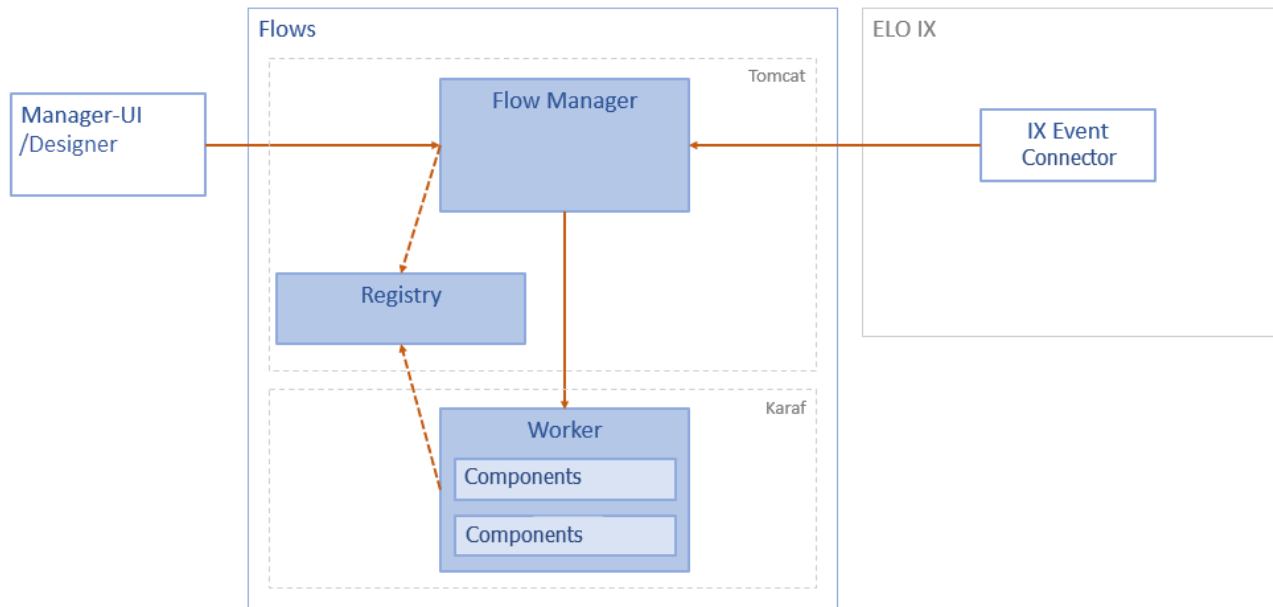
You will find more information about the following in the *Installation & operation* section:

- Architecture
- Installation and operation

You will also find tips for troubleshooting and collecting error data to speed up analysis in the ELO Community or by Business Support.

Architecture

The architecture of ELO Flows provides a web-based front end and a scalable worker model. ELO Flows also enables component management with hot deployment support through the integration of Apache Karaf.



Modules

ELO Flows is made up of different modules.

The ELO Flows Manager and the registry run on a Tomcat as a web app (as do the other ELO modules). The workers run on an [Apache Karaf](#).

ELO Flows Manager

The ELO Flows Manager is a main instance in ELO Flows. It is responsible for starting new flows and managing existing ones.

The ELO Flows Manager provides the interface for flow administration. Flows can be created and managed via this interface. The interface can also access registry information.

Flow administration

The flow administration area is the interface in the ELO Administration Console. You can manage flows here. The app is provided together with the required interfaces by the ELO Flows Manager.

Workers

The workers are independent containers. Components are deployed in the workers. Multiple containers can be used in parallel for load balancing purposes.

Registry

The registry is a main instance in ELO Flows. The workers authenticate with the registry and register their components there. The registry also checks whether the workers can be reached and is notified if the components of a worker change.

ELOix event connector (OSGi plug-in)

The ELOix plug-in is responsible for forwarding ELOix events to the ELO Flows Manager.

Currently, a differentiation is made between the following events:

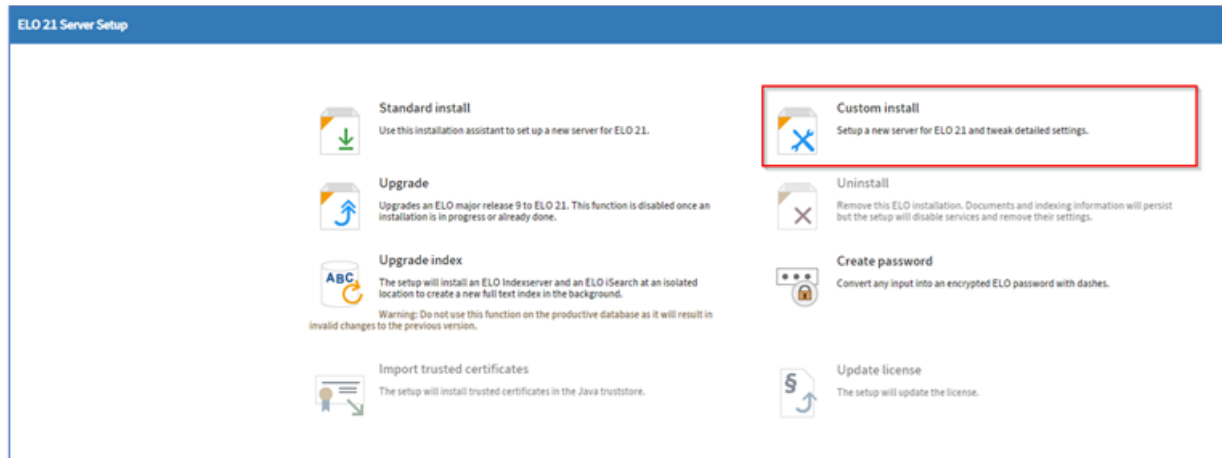
- Creation of a SORD
- Update of an existing SORD
- Call of a dynamic keyword list from a form

Components

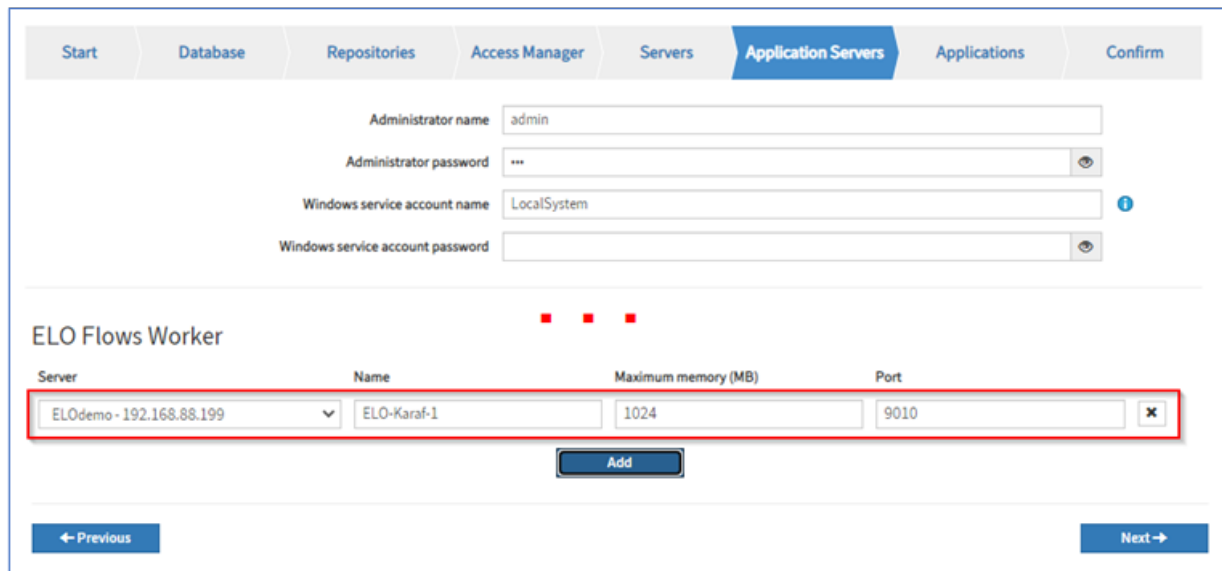
The components are developed as OSGi plug-ins. They provide triggers and services.

Installation

1. Start the ELO Server Setup.



2. Select the option *Custom install*.



3. On the *Application Servers* page, add an entry for the ELO Flows worker with *Add*.

Type	ELO Server Engine
ELO Administration Console	ELO-ELOdemo-2 @ ELOdemo
ELO OCR Service	ELO-ELOdemo-2 @ ELOdemo
ELO Flows Engine	ELO-ELOdemo-1 @ ELOdemo
ELO Flows Registry	ELO-ELOdemo-1 @ ELOdemo

[Add OCR](#) [Add ELO Interface for Microsoft Office Online](#) [Add ELO Replication](#) [Add ELO Flows Engine](#) [Add ELO Flows Registry](#)

4. On the *Applications* page, add the two ELO Flows server applications *ELO Flows Engine* and *ELO Flows Registry*.
5. Finish the ELO Server Setup.

Manager authentication

After installation, the flow administration area can be opened in the ELO Administration Console.

Alternatively, you can open it directly via a URL structured as follows:

`http://<HOSTNAME>:<PORT>/ix-<REPOSITORYNAME>/plugin/de.elo.ix.plugin.proxy/flows/`

Troubleshooting

Troubleshooting

Troubleshooting can be performed in multiple steps:

1. Check whether all modules are running properly. Multiple status pages are available here.

The Monitoring section explains how to reach them.

2. Check the log files.

You will find the log files *worker.log* and *manager.log* in the flows working directory. Errors begin with the prefix ERROR.

With the complete error stack trace, you can check the cause for the error or share it with support, development, or the community.

Monitoring

ELO Flows provides a monitoring tool. In this tool, you can monitor the processing of flows and analyze errors.

 Icons

 Transformations



1. Select *Monitoring* in the flow administration area.

Status Report

RUNNING

Version Messages Flows Flows timeline

Cache

Refresh Cache

The *Status Report* page opens. The page consists of the following tabs:

-

Version: Shows the version of ELO Flows currently being used

- Messages: Shows current status messages
- Flows: Shows the available flows with the corresponding statistics
- Flows timeline: Shows the timeline of executed flows and, if applicable, errors in the list. The list can be updated, searched, and filtered
- Validation errors: Shows any errors and error sources in flows and transformations
- Controls: When you select the *Refresh Cache* button, the ELO Flows Manager reads the database contents again

Alternatively, you can perform manual troubleshooting, supported by the following status pages.

Status pages

Multiple status pages are available, which you can check. In addition to the currently installed version number, they also indicate the online status of the modules:

Simple status page of the ELO Flows Manager

ELO Flows Status Report

Running

The manager offers a very simple status page that only shows the status and can be used without authenticating:

```
http://<hostname>:<port>/flows/status
```

Flow administration status page

Status Report

RUNNING

Version Messages Flows Flows timeline

Cache

Refresh Cache

You can reach the flow administration status page at the following URL:

```
http://<hostname>:<port>/ix-<repository>/plugin/de.elo.ix.plugin.proxy/flows/#/status
```

ELO Flows registry status page

ELO Flows Registry

Status Report

RUNNING

Version Services Messages

Version

23.00.000.5108-47a22080

You can reach the ELO Flows registry status page at the following URL:

```
http://<hostname>:<port>/registry
```

Information

Change the host name and port based on the respective installation.

Known issues

Transfer ELO permissions

Output ACL objects are not currently compatible with input ACL objects. To transfer permissions, they have to be mapped. This requires a transformation.

Workaround



ELO objects and metadata
com.elo.flows/ELOBase/v1



Add permissions

Information

Object ID* ✓ ab J: ▾

ACL inheritance mode* ab J: ▾

Permissions List of permission entries Array

Input mode* ▾

Permissions*

1	<code>\$map(step0.acl,function(\$v){{"user": \$v.user.id,"read": \$v.read,"write": \$v.write,"delete": \$v.delete,"edit": \$v.edit,"list":\$v.list,"permission":\$v.permission}})</code>	✓	ab J: ▾
---	--	---	---------

1. Create the service *Add permission*.
2. Navigate to the *Settings* tab.
3. Enter the *Object ID*.
4. Select an *ACL inheritance mode*.
5. Open the list settings menu for permissions by selecting *Array*.
6. Select *Basic entry* as the *input mode*.
7. Transfer the following JSONata call in the JSONata editor under *Permissions*:

```
$map(stepX.acl, function($aclItem) {
  {
    "user": $aclItem.user.id,
    "read": $aclItem.read,
    "write": $aclItem.write,
    "delete": $aclItem.delete,
    "edit": $aclItem.edit,
    "list": $aclItem.list,
```

```
"permission": $aclItem.permission
}
})
```

8. Adapt the context to the source object of the permissions.

Handling of date values with the ELO Flows JSONata editor

Internally, ELO uses the ISO format without a delimiter (e.g. 20130701).

The `$toMillis()` function in ELO Flows does not currently support conversion without a delimiter.

Workaround

Delimiters must therefore first be inserted in the ISO date.

`$milliRes` is the date converted into milliseconds which you can then use to continue working.

```
(
  $iDate := activities.calendar.date;
  $cDate := $substring($iDate,0,4) & "-" & $substring($iDate,4,2)
           & "-" & $substring($iDate,6,2);
  $milliRes := $toMillis($cDate);
)
```

Concatenation in transformation fields

ELO Flows enables concatenation within transformation fields, for example in the *List transformation* field.

When making an entry in the JSONata editor, place your expression in brackets. Without the brackets, the concatenation is applied to the entire expression instead of to the individual elements of the array.

Please note

Concatenation in transformation fields is only possible in expert mode in the JSONata editor. Select the mode with the *J*: icon.

Example concatenation of GUID and string

Incomplete Finish Properties

Flow properties Versions Save Create version

[trigger] Dynamic keyword list called

Object properties

Title of table*

Columns The title of the table Array

Input mode*

Context* 1 ab J: v

Column title* ab J: v

Form field name* ab J: v

Column data The column data Array

Input mode* List transformation v

Context* 1 ab J: v

string 1 (\$.guid&"HelloWorld") ab J: v

```
/*Input in JSONata editor*/

($.guid&"HelloWorld")

/*Result of the concatenation*/

[
  "guid1HelloWorld",
  "guid2HelloWorld",
  "guid3HelloWorld"
]
```

Migrate packages from old data

The ELO Flows migration is carried out once. A flag is then set in the database indicating that the migration of old data has already been performed. Migration is not started over for this reason. Loading non-migrated data permanently destroys it.

There isn't currently an event for package import able to correctly handle the imported old data. Use either your own import from ELO Flows (old flows are correctly migrated here) or the workaround described below.

Workaround

1. Stop the ELO Flows Manager service.
- 2.

Delete the *migration flag* in the database.

Example:

```
delete from configurations where cfgcomponent = 'flows'  
and cfgkey = '23.00.000'  
and cfggroup = 'TEMPLATE_MIGRATION'
```

3. Start the ELO Flows Manager service.

Result: Old data is migrated again the next time ELO Flows starts.

Problems in the ELO Java Client when running manual flows without activities

Due to cookie guidelines, it isn't possible to execute manual flows without activities in the ELO Java Client if they are running on the same machine as the ELO Flows Manager. The reason for this is that the ELO Flows Manager is unable to verify the request if the client doesn't transfer a JSESSION-ID. Calls of manual flows with activities, however, always work, as they are executed in the client's web view. The ELO Web Client is not affected by the problem.

Use

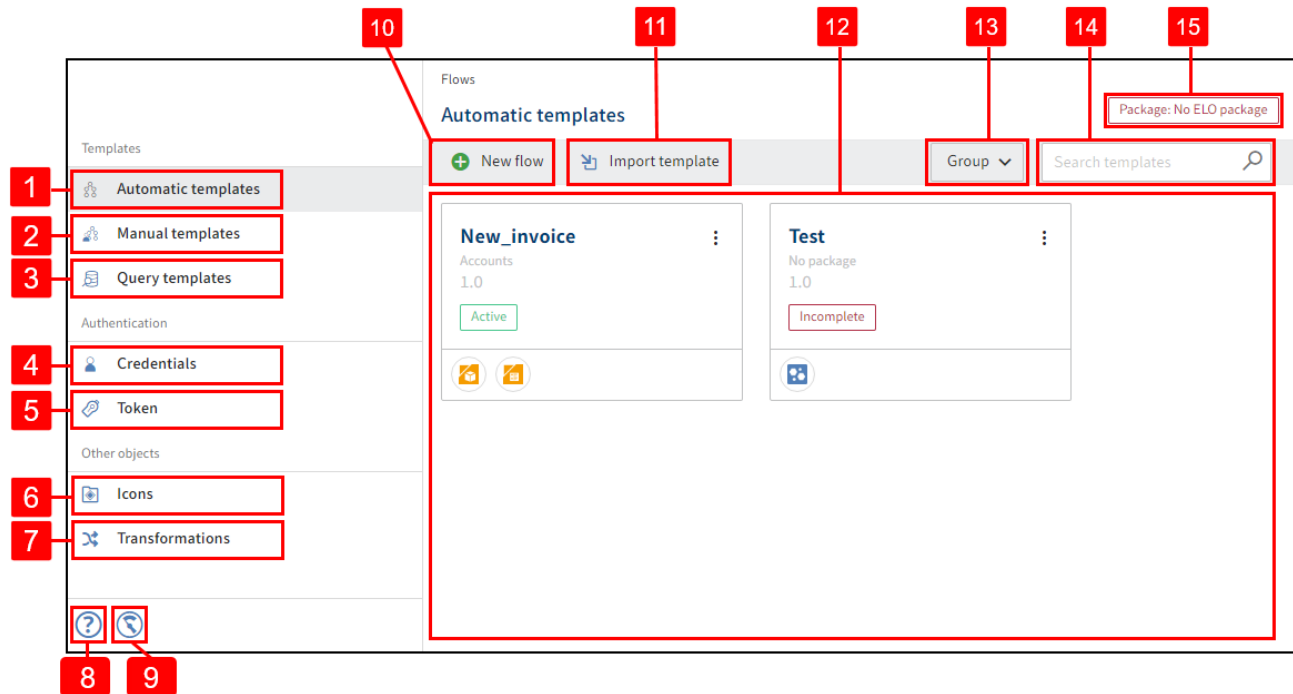
Flow administration

The flow administration area is the central application for managing ELO Flows. You can reach it in the [ELO packages](#) context.

Information

Flows can only be created, configured, or started by users with the "Main administrator" role.

Home screen



1 Automations: On start-up, the *Automations* menu item is selected. You can create flows with automatic triggers via this menu item.

You can find more information under Automations.

2 User actions: Selecting the *User actions* menu item shows you the current flows with manual triggers.

You will find more information under User actions.

3 Data query: With the *Data query* menu item, you can create special flows that use triggers in the form of queries.

4 Credentials: Under *Credentials*, you can configure credentials for each component.

You will find more information under Credentials.

5 Token: With the *Token* menu item, you create tokens. You can safeguard trigger events that are triggered by third-party system calls by creating a token with an assigned user context.

You will find more information under Webhooks and token authentication.

6 Icons: Select the *Icons* menu item to view existing icons and create new ones.

You will find more information under Icons.

7 Transformations: You can create transformations via the *Transformations* menu item.

Refer to Transformations

for more information.

8 Help: Opens the help page in a new tab.

9 Monitoring: Select *Monitoring* to open the *Status Report* page. There, you will find information about the status of the module and executed flows.

10 New flow: Create a new user action, automation, or data query.

11 Import: Import exported flows as a JSON file.

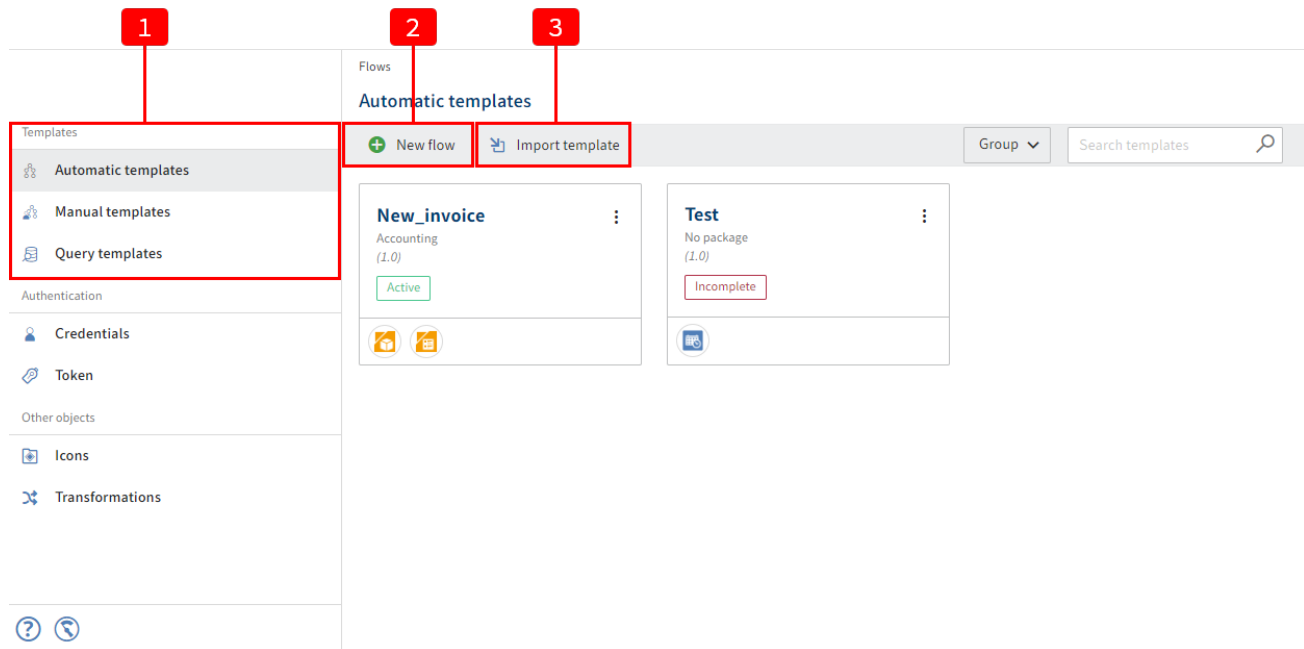
12 Viewer pane: Here you can see the flows currently available.

13 Group: You can sort the available templates via the *Group* drop-down menu.

14 Search templates

15 Package: Here, you see whether the current flows are part of a package in ELO.

Create new flow



1. In the sidebar, select a type for your flow (1) .
2. Depending on the flow type, select *New automation*, *New user action*, or *New data query* (2).
Alternative: By selecting *Import* (3), you can import exported flows as a JSON file.
3. Enter the corresponding metadata for the flow to be created:

The screenshot shows a 'Flow properties' dialog box with the following elements:

- 1**: A red box around the 'Create' button.
- 2**: A red box around the 'Cancel' button.
- 3**: A red box around the 'Identifier*' text input field, which contains a vertical cursor and the text 'Input required.' below it.
- 4**: A red box around the 'Package' dropdown menu, which shows 'Select package' and a downward arrow.
- 5**: A red box around the 'Description' text area.

1 Create: Saves the entries. The new flow is created.

2 Discard: Cancels flow creation.

3 Name: Unique, technical name of the flow.

Information

Enter a name for the flow that briefly describes what the flow does.

Information

Do not use special characters.

The following characters are allowed:

- a-z
- A-Z
- 0-9
- äöü
- ÄÖÜ
- ß
- -

4 Package: Shows which package context the flow is created in.

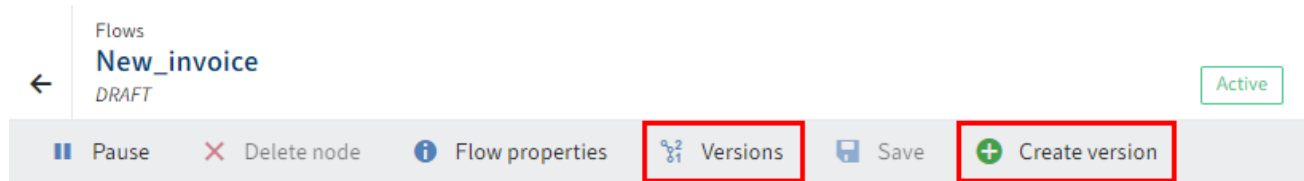
5 Description: Detailed description of the function of the flow. Use this field if your flow is complex.

4. Select *Create*.

You will see the new flow in the overview. Initially, it will show *Incomplete*, as no triggers or services have been defined yet.

Flow versions

In the flow designer, you can save different versions of your flows.



Create version

Create a backup of the current version of your flow with the *Create version* function.

1. Edit a flow in the flow designer.
2. Save your version by selecting *Save*.

Information

Flows
⚙️ ? 📄

Versions
✕

Version	Comment	Status	Last changed on
DRAFT		Incomplete	


When you select *Save*, ELO Flows automatically generates the *DRAFT* version. You can find this draft version in the *Versions* overview.

Every time you save, the *DRAFT* version is updated and overwritten.

3. To back up the current version, select *Create version*.

Versions

Create version ✕

 Create

Version* ✓

Comment ✓

26 / 1024

The *Create version* window opens.

4. Enter a name for your version of the flow.

Optional: Comment on the version.

5. Select *Create* next to the floppy disk icon.

Result

Flows

Versions ✕

Version	Comment	Status	Last changed on
Basic_version	Basic trigger and service	Active	

The newly created version is now available in the *Versions* overview. There, you can select the individual versions to continue processing them and activate/deactivate them in the flow designer, for example.

Information

Flows

Versions ✕

Version	Comment	Status	Last changed on
Version2		Active	11/18/2022
Basic_version	Basic trigger and service	Inactive	11/18/2022 🗑️

The new version assumes the status of the previous DRAFT version. If the DRAFT is active, the new version is automatically activated. If the DRAFT is deactivated, the new version is automatically deactivated.

Select version and continue processing

You can select created versions individually to continue processing the flow based on them.

1. Select *Versions*.

The *Versions* window opens.

2. Select the version you would like to process.

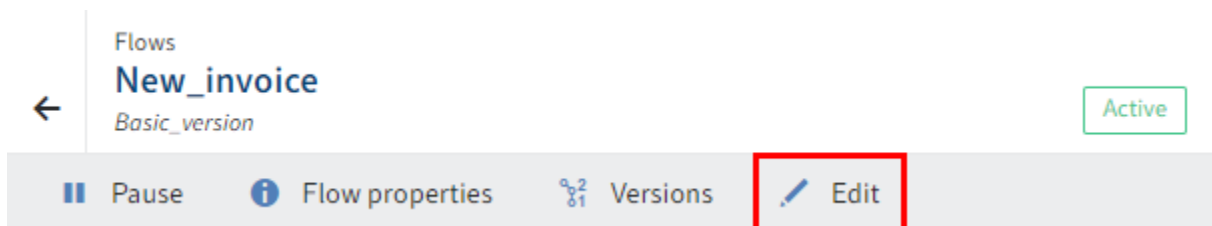
If the desired version is already selected, you can close the window by clicking the X icon.

Result

The *Versions* window closes. You are back in the flow designer.

Edit version

Using the *Edit* function, you can modify a flow based on the current status of a version.





1. Select the version in the *Versions* window.
2. Select *Edit*.
3. Edit the flow.
4. Save your draft version.

Result

Flows

Versions ✕

Version	Comment	Status	Last changed on	
Version1	Basic version	Inactive	11/18/2022	
Version2	Service added: "External link"	Inactive	11/18/2022	
DRAFT		Active	11/18/2022	

The version you used to edit the flow remains unchanged. You can find the current status as a *DRAFT* version in the *Versions* window.

Outlook

Create a new version to save the current status as a version.

Delete version

To delete a version, it must first be deactivated.

1. Select the version to delete in the *Versions* window.
2. Select *Pause* in the flow designer.
3. Select *Versions*.
4. Select the recycling bin icon beside the version to delete it.

Delete flow template



Do you really want to delete the template?

Confirm deletion of the template.

Version2

Cancel

Delete template

5.

Confirm deletion in the *Delete* dialog box.

Result

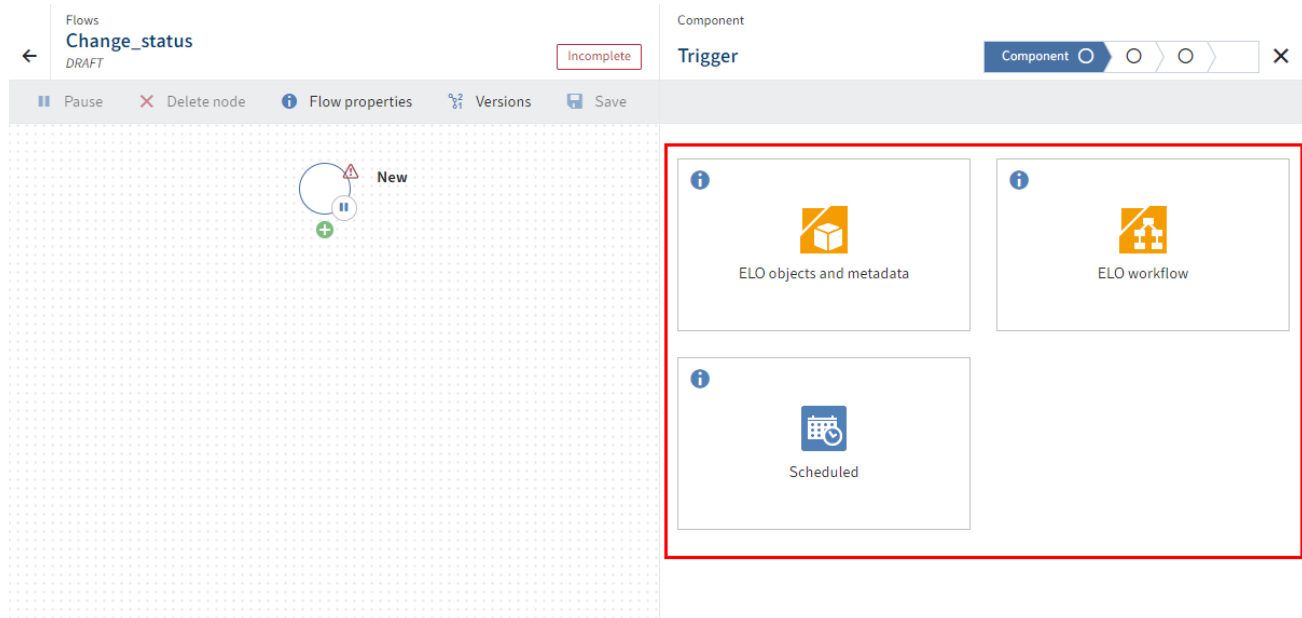
The version you selected is deleted and is no longer shown in the *Versions* window.

Automations

Flows in the *Automations* area consist of an (automatic) trigger and of at least one or more services. Optionally, you can make flows more complex by using branches.

Select trigger

First, select a trigger. Proceed as follows:



1. Select a *component* whose trigger you want to use.


The flow designer automatically jumps to the following *Trigger* tab.


2. Select the *trigger* you want to start this flow.

The flow designer automatically jumps to the following *Settings* tab. The setting options for the trigger are shown here:

Settings

Trigger ✓ ✓ Settings ○ ✕

 ELO objects and metadata
com.elo.flows/ELOBase/v1

 Object edited

ℹ Information

Form ℹ

Field monitoring Array

Next

In this example, you can restrict the *Object changed* trigger to specific metadata forms by entering the metadata form name. Other triggers have other settings here.

Information

Under Components > General information, you will find more information on completing fields with gen. 1 and gen. 2 metadata in the flow designer.

3. Select *Next* to open the *Summary* tab.

Summary

Trigger Component ✓ Trigger ✓ Settings ✓ **Summary** ✕


▼ Properties

ID


1 Node key*

2 Node description ✓

3 ▼ Component

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

4 ▼ Service

 **Object edited** (ELO trigger)

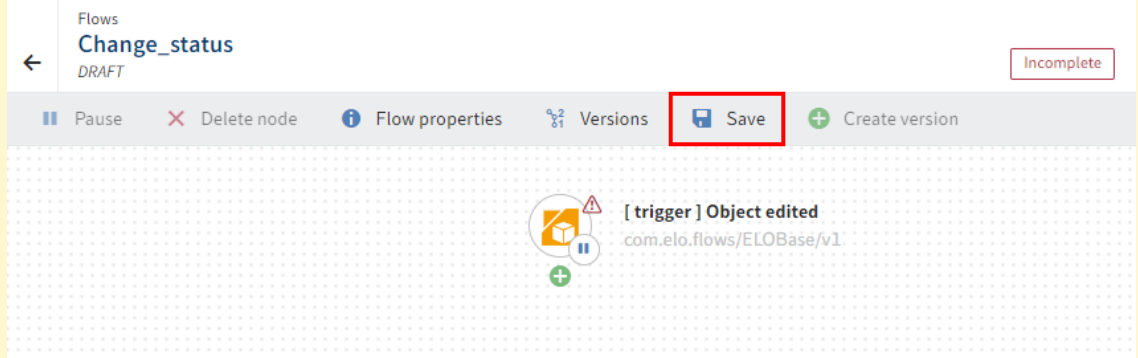
5 ▼ Settings

Node key

This tab shows a summary of all settings:

- 1 Node key: Name of the node shown in square brackets in the flow designer
 - 2 Node description: More detailed description of the node
 - 3 Component: Displays the selected component
 - 4 Service: Displays the selected trigger with a copyable path to the endpoint
 - 5 Settings: Displays the node key
4. Select *Done* to finish configuring the trigger.

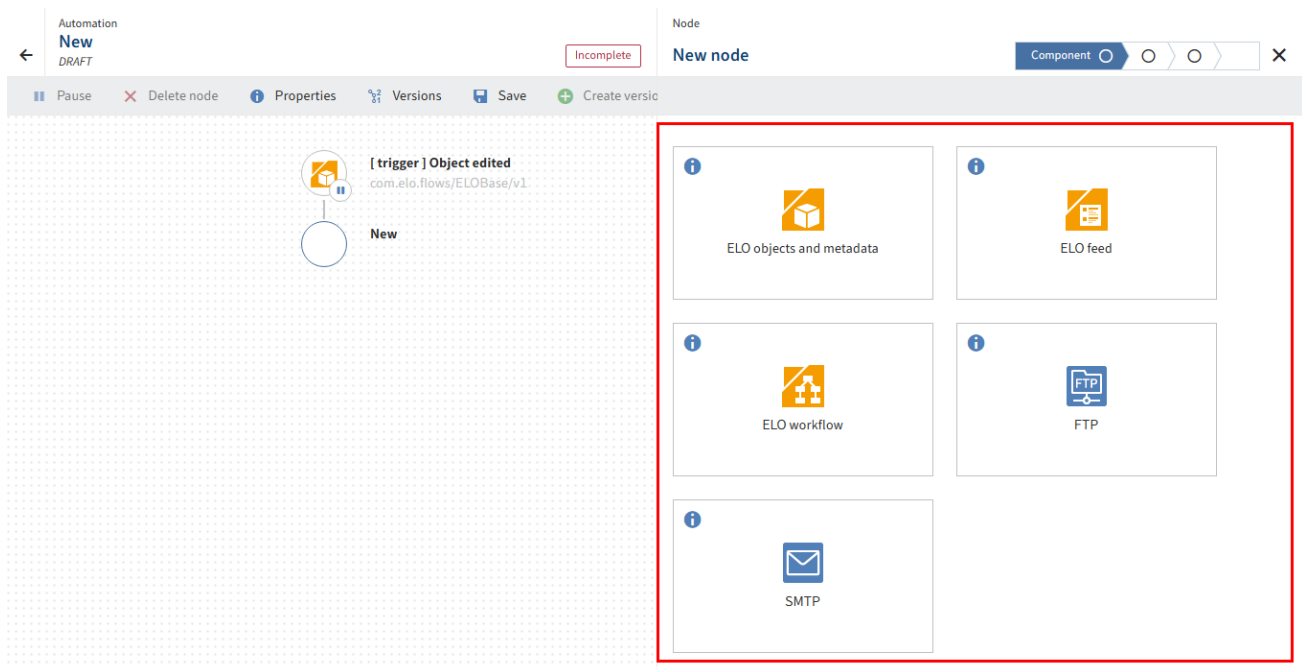
Please note



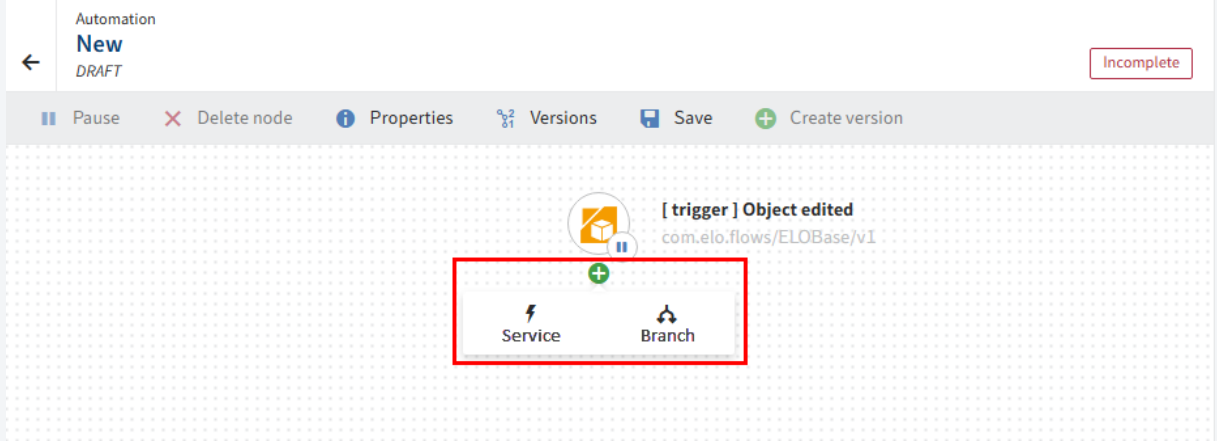
With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

Add service



Information



Before each step, you can decide whether to create a branch with a true-false condition or to add a service. You can nest branches in other branches or continue with services.

You will find more information on branches in ELO Flows in this documentation under Advanced flow.

1. To add a service, select the round plus icon next to the trigger node and select *Service*.

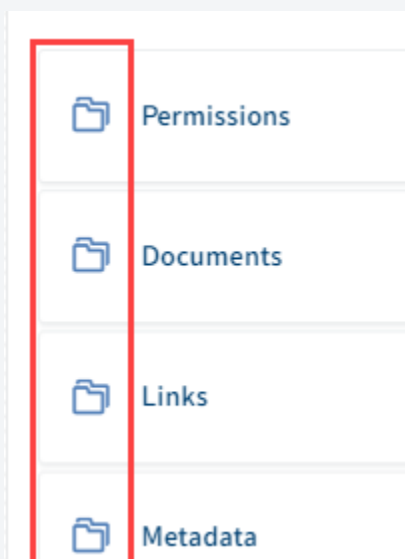
The configuration area with the available components opens for the new node.

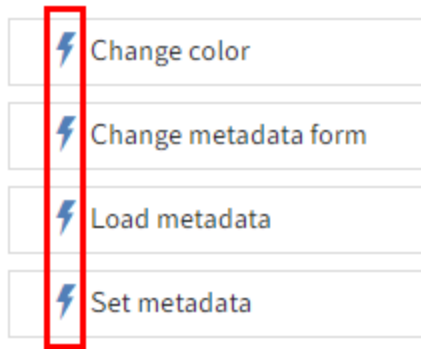
2. Select a component.

By selecting the component, you are automatically taken to the next tab with the services available for the component.

3. Select the desired service.

Information

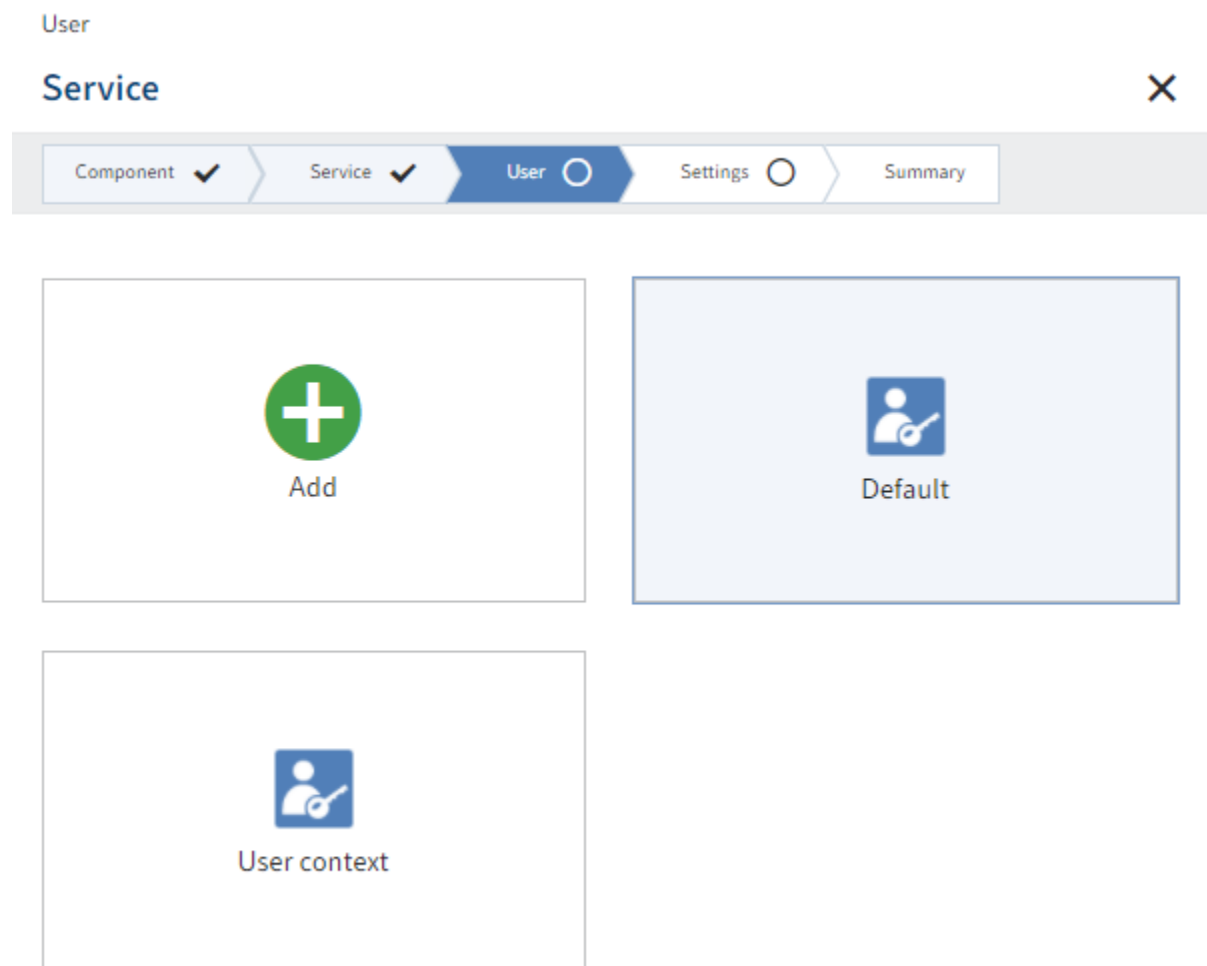




Services can be listed individually (lightning icon) or in groups (folder icon).

Selecting the service automatically takes you to the next tab with the user settings.

4. Select a connection.



Information


If you've already configured a user under Credentials, you can select the user here. Otherwise, you can select the default connection or add a new one.


5. Confirm your selection with *Next*.

The *Settings* tab appears.

Settings

Service Component ✓ Service ✓ User ✓ **Settings** Summary ✕

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

 Change color

i Information

Object ID* **i**

Color **i**

Next

6. Configure custom data for the service.

Information

You can complete the fields with *static* or *dynamic values*. Dynamic evaluation is performed based on *JSONata*. With this transformation language, you can access event objects of the previous nodes and triggers in addition to using transformations or simple functions.

You will find more about the use of *JSONata* in ELO Flows under *JSONata* editor.

- 7.

Confirm your selection with *Next*.


Information


Some services offer you the output schema function. That means that this service offers a dynamic output object. This can be the case with metadata objects, for example.

Example: You file a document with the *E-mail* metadata form and want to access a field of this document in a subsequent node. As ELO Flows initially does not know which fields are relevant for this document, the service only offers the parent object (e.g. `objKeys` and `aspectData`) in the subsequent object without access to the individual metadata fields.

Output schema

Service ✓ ✓ ✓ ✓ Output schema ○ ×

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

 Load metadata

Information

▼ **aspectData**
Metadata form

Use default Load schema

▼ **objKeys**
Metadata form


Use default Load schema


So that ELO Flows knows which metadata fields the output object has, you can either use a *standard metadata form* or load a special one. In the metadata form field, enter


E-mail and select *Load schema*. The schema is loaded with the appropriate metadata form properties.

Output schema

Service Component ✓ Service ✓ User ✓ Settings ✓ **Output schema** Summary ✕

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

 Load metadata

 Information

▼ objKeys

Schema

```
{
  "type": "object",
  "title": "elobasecomponent.objkeys.title",
  "description": "elobasecomponent.objkeys.descr"
}
```

Reset schema

▼ aspectData

Schema

```
{
  "type": "object",
  "title": "elobasecomponent.aspectdata.title",
  "description": "elobasecomponent.aspectdata.descr"
}
```

Next

Alternative: You can also select *Use default*. The object is then output with the default properties, and the properties of the object have to be accessed manually via JSONata.

If applicable, confirm the settings on the *Output schema* page by clicking *Next*.

Summary

Service

Component ✓ Service ✓ User ✓ Settings ✓ Summary X

▼ Properties


ID d4125596-27ad-4f6e-8dd8-680c71884e4b

1 Node key* [...] step0

2 Node description

3 Do not trigger an error if the service fails.

▼ Component

 ELO objects and metadata
com.elo.flows/ELOBase/v1

Done

The *Summary* tab opens. All the settings are shown once again.

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

3 Do not trigger an error if the service fails.: The entire flow runs, even if this service fails. This function is for test purposes. The result data for this service may be faulty or unavailable.

8. Select *Done* to apply the service settings in the configuration area.

Please note

With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

9. Save the entire flow.

The status display now changes to *Active*.

User actions

Flows in the *user actions* area consist of a manual trigger and of at least one or more services. Optionally, you can make flows more complex by using branches.

Select trigger

For user actions, you don't select a trigger for a component, but instead define a button for ELO clients. This button starts the flow.

1. Open the trigger in the node editor by selecting the trigger node.

The *Action definition* tab opens.

You have the following configuration options:

- 1 Select label: Button title shown in the ELO clients
- 2 Add tip: Tooltip text for the button you've created

3 Configure metadata form: Configure the relevant metadata form for this button. The button is only enabled if the selected element has the defined metadata form.

Information

Under Components > General information, you will find more information on completing fields with gen. 1 and gen. 2 metadata in the flow designer.

4 Enable context for...: This button is only enabled if the selected element is a folder and/or a document. After activation, the object ID of the object selected in ELO Flows is provided.

5 Show as a tile: The user action is shown as a tile in the *My ELO* work area. You can configure an icon for it.

6 Show on ribbon: Defines the position of the button on the ribbon in the ELO clients. Select the *tab*, the *group*, and the *icon*.

7 Only show in certain ELO clients: Controls which clients the button is shown in. If nothing is selected here, the button is shown in all clients.

8 Users/groups: Defines which users and groups the button is shown to. If this option is not enabled, the button is shown to all users.

2. Complete at least the mandatory fields.

3. Select *Next* to confirm your entries.

Node

Manual trigger

✓ Summary X

▼ Properties

ID c64b9aaf-eac4-4b21-972b-e19f19394f50

1 Node key* [...] trigger


2 Node description

3 Enable synchronous feedback

▼ Action definition

Label
TREE.STUDENTS

▼ Service

 Manual

▼ Settings

Node key
trigger (c64b9aaf-eac4-4b21-972b-e19f19394f50)

Done

The *Summary* tab opens. All the settings are shown once again.

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

3 Enable synchronous feedback: Executes the flow as a synchronous flow

Information

You can find more information on this topic in the Synchronous flows section.

4. Select *Done* to finish configuring the trigger.

Please note

With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

Add activities

Via *Add activity*, define what to show when the button is selected in the client.

Information

Activities are optional.

The screenshot shows the 'Flow Designer' interface for a flow named 'Order' (DRAFT). The main canvas displays a flow diagram with a trigger node and a plus icon next to it. The 'Add activity' panel on the right lists various activity types: Information, Agreement, Multiple docum..., Document upload, Date picker, Simple form, Form, Summary, Repository sele..., and User selection.

1. To add an activity, select the plus icon next to the trigger node.

All activities currently available are shown on the *Activity* page.

- Information: Create a purely informational text that is shown in the client. This activity does not provide any data.
- Agreement: Define an agreement shown in the client via a check box. A data privacy agreement, license agreement, etc. can be requested in this way, for example.
- Multiple document upload: With this activity, you can prompt the user to upload multiple documents. You can save them in ELO with the ELO objects and metadata, for example.
-

Document upload: Request a document to be uploaded in the client. You can save it in ELO with the ELO objects and metadata, for example.

- Date picker: Shows a calendar. Users can select a specific date or date interval here.
- Simple form: Create a form which is then shown in the client. For example, you can implement a contact form with this activity.
- Form: Select a preconfigured form from the package context. You can also check the following boxes: *Load context in this form* and *Use query flow*.
- Summary: The summary shows all data from previous activities. You can place this at the end to give users a final overview.
- Repository selection: With the *Repository selection* activity, you can prompt users to select an element from the ELO repository. In the settings, set the *root element* to define the level at which users are offered a selection.
- User selection: With this activity, you can prompt users to select an ELO user and/or ELO group from the ELO system.

Information

You can select localization keys for activities with text or titles visible to the user in the client.

If you configure flows outside of ELO packages, you can only use keys available in the system.

The keys can come from the following sources:

- Keys already configured via ELO packages
- Keys filed in the repository as properties files

For flows you edit within ELO packages, you can also create new localization keys and configure existing ones. The localization is configured for the active language of the ELO Administration Console.

1. Enter a localization key.

Alternative: Select an existing localization key from the drop-down menu.

2. Enter a name in the *Localization* field.

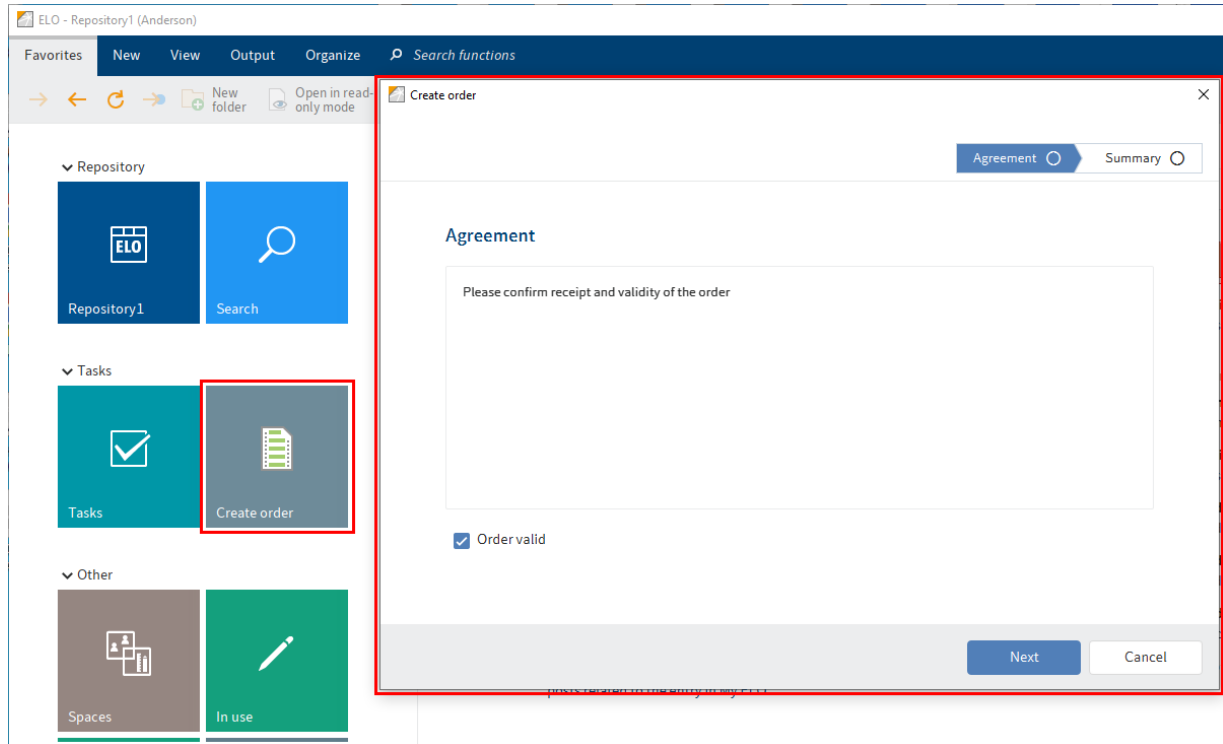
3. Select the floppy disk icon to assign the name to the localization and save it.

2. Select an activity.

Option 1: To map a series of activities, these can be chained. Add another activity.

Option 2: Delete the existing activity by selecting *Delete node* on the toolbar.

The button and activities in the ELO client can be shown as follows:

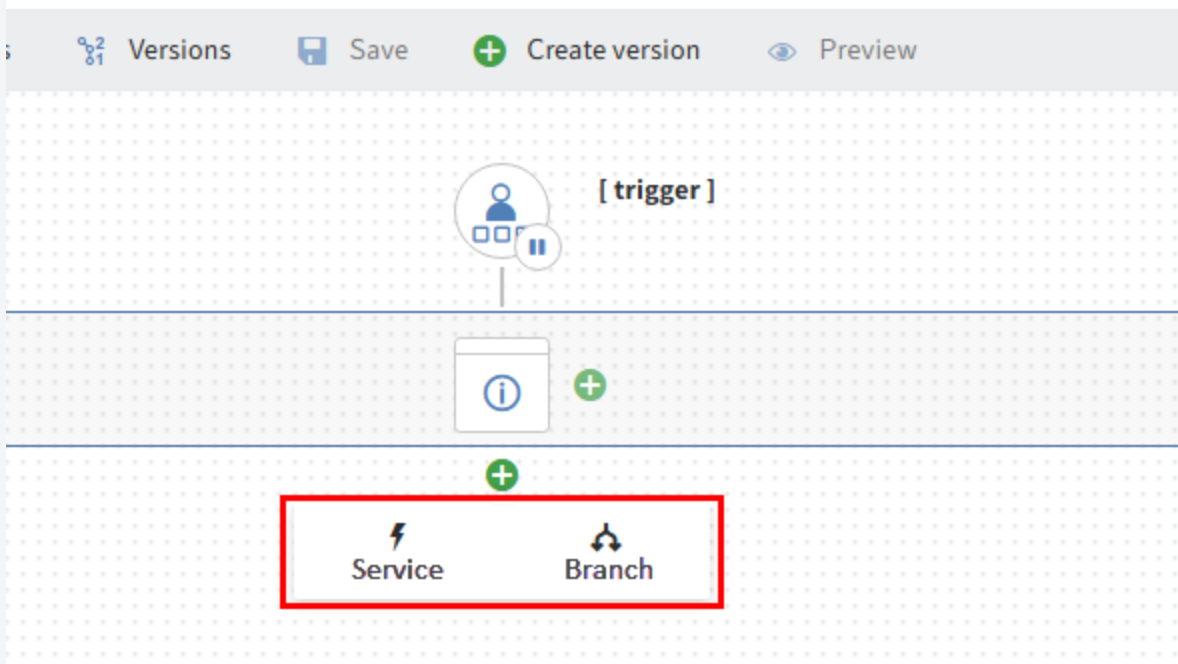


Please note

The button is only shown if the status of the flow is *Active*. You have to define at least one service.

Once you have defined activities, you can create your first branch or service.

Information



Before each step, you can decide whether to create a branch with a true-false condition or to add a service. You can nest branches in other branches or continue with services.

You will find more information on branches in ELO Flows in this documentation under Advanced flow.

Add service

The screenshot shows the ELO Flow editor interface with the 'Add service' configuration panel open. The panel is titled 'Component Service' and has a 'Component' dropdown menu. Below the title, there is a toolbar with icons for 'Flow properties', 'Save', and 'Preview'. The main area of the panel displays a list of available components, each with an icon and a label: 'ELO objects ...', 'ELO feed', 'ELO workflow', and 'SMTP'. A red rectangular box highlights this list of components. In the background, the flow diagram shows a trigger node labeled '[trigger]' and a new node labeled 'New' with a plus sign.

1. To add a service, select the plus icon next to the trigger node and select *Service*.

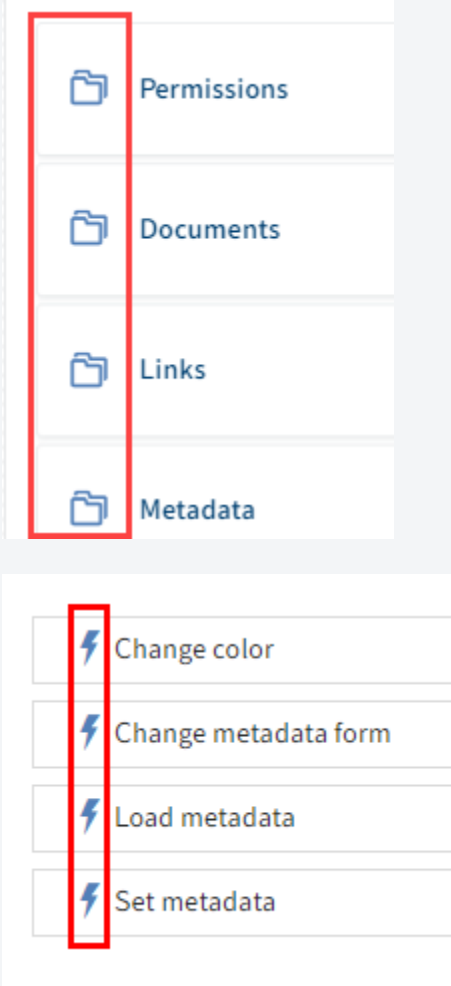
The configuration area with the available components opens for the new node.

2.

Select a component.

By selecting the component, you are automatically taken to the next tab with the services available for the component.

3. Select the desired service.



Information

- Permissions
- Documents
- Links
- Metadata

Change color

Change metadata form

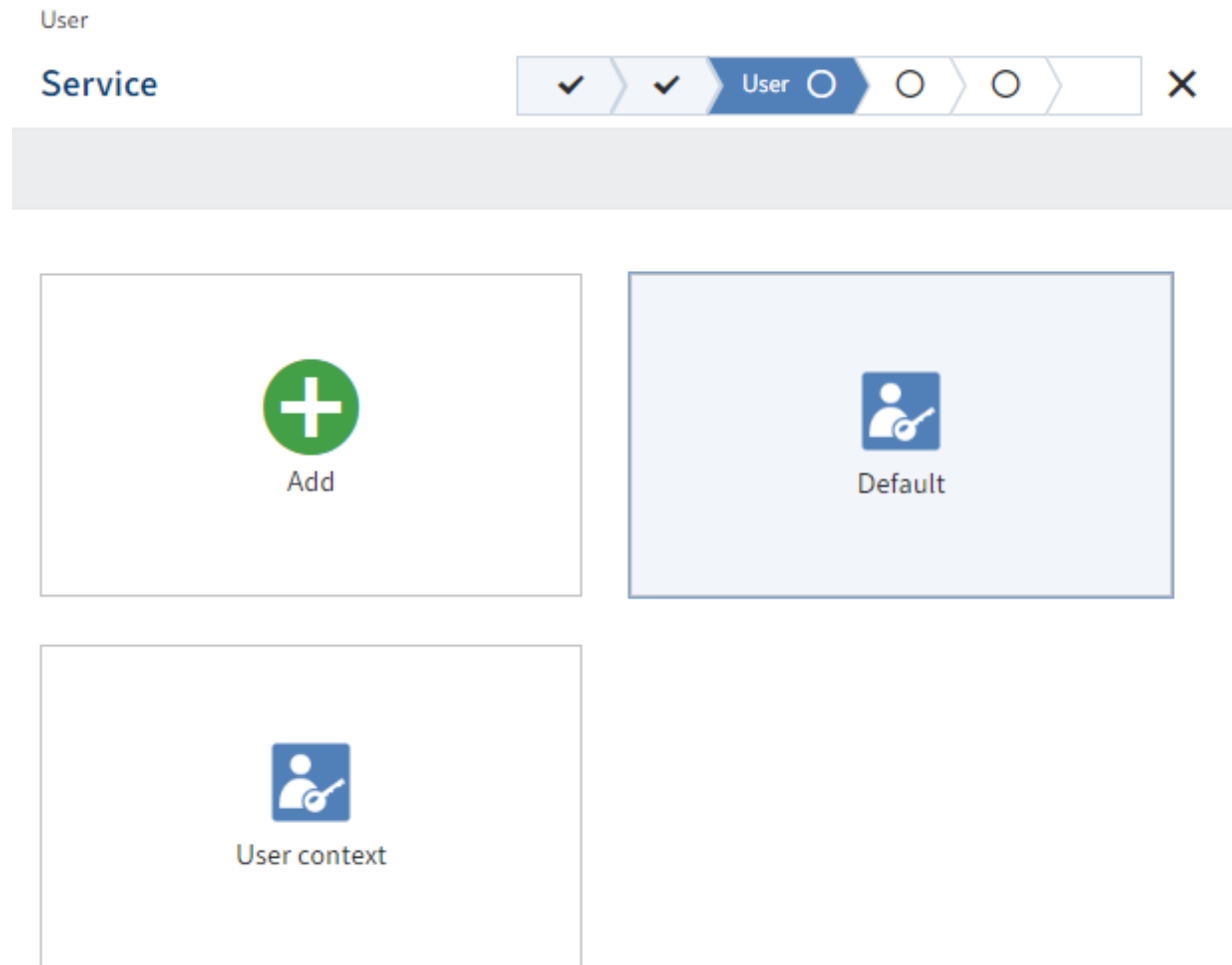
Load metadata

Set metadata

Services can be listed individually (lightning icon) or in groups (folder icon).

Selecting the service automatically takes you to the next tab with the user settings.

4. Select a user connection.

**Information**


If you've already configured a user under Credentials, you can select the user here. Otherwise, you can select the default connection or add a new one.


5. Confirm your selection with *Next*.

The *Settings* tab appears.

Node

New node ✓ ✓ ✓ **Settings** ✓ ✕

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

 Change color

i Information

Object ID* i ab J: ▼

Color i ab J: ▼

Next

6. Configure custom data for the service.

Information

You can complete the fields with *static* or *dynamic values*. Dynamic evaluation is performed based on *JSONata*. With this transformation language, you can access event objects of the previous nodes and triggers in addition to using transformations or simple functions.

You will find more about the use of *JSONata* in ELO Flows under *JSONata* editor.


Information


Some services offer you the output schema function. That means that this service offers a dynamic output object. This can be the case with metadata objects, for example.


Example: You file a document with the *E-mail* metadata form and want to access a field of this document in a subsequent node. As ELO Flows initially does not know which fields are relevant for this document, the service only offers the parent object (e.g. objKeys and aspectData) in the subsequent object without access to the individual metadata fields.

Output schema

Service ✓ ✓ ✓ ✓ Output schema ○ ✕

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

 Load metadata

 Information


▼ **aspectData**
Metadata form


▼ **objKeys**
Metadata form


So that ELO Flows knows which metadata fields the output object has, you can either use a *standard metadata form* or load a special one. In the metadata form field, enter *E-mail* and select *Load schema*. The schema is loaded with the appropriate metadata form properties.

Output schema

Service Component ✓ Service ✓ User ✓ Settings ✓ Output schema ○ Summary ✕

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

 Load metadata

 Information

▼ objKeys

Schema

```
{
  "type": "object",
  "title": "elobasecomponent.objkeys.title",
  "description": "elobasecomponent.objkeys.descr"
}
```

Reset schema

▼ aspectData

Schema

```
{
  "type": "object",
  "title": "elobasecomponent.aspectdata.title",
  "description": "elobasecomponent.aspectdata.descr"
}
```

Next

Alternative: You can also select *Use default*. The object is then output with the default properties, and the properties of the object have to be accessed manually via JSONata.

If applicable, confirm the settings on the *Output schema* page by clicking *Next*.

Summary

Service

Component ✓ Service ✓ User ✓ Settings ✓ Output schema ✓ Summary X

▼ Properties


ID 780517da-8e65-45ff-a2a9-7ad1d6010f41

1 Node key* [...] step0

2 Node description

3 Do not trigger an error if the service fails.

▼ Component

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

Done

The *Summary* tab opens. All the settings are shown once again.

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

3 Do not trigger an error if the service fails.: The entire flow runs, even if this service fails. This function is for test purposes. The result data for this service may be faulty or unavailable.

7. Select *Done* to apply the service settings in the configuration area.

Please note

With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

8. Save the entire flow.

The status display now changes to *Active*.

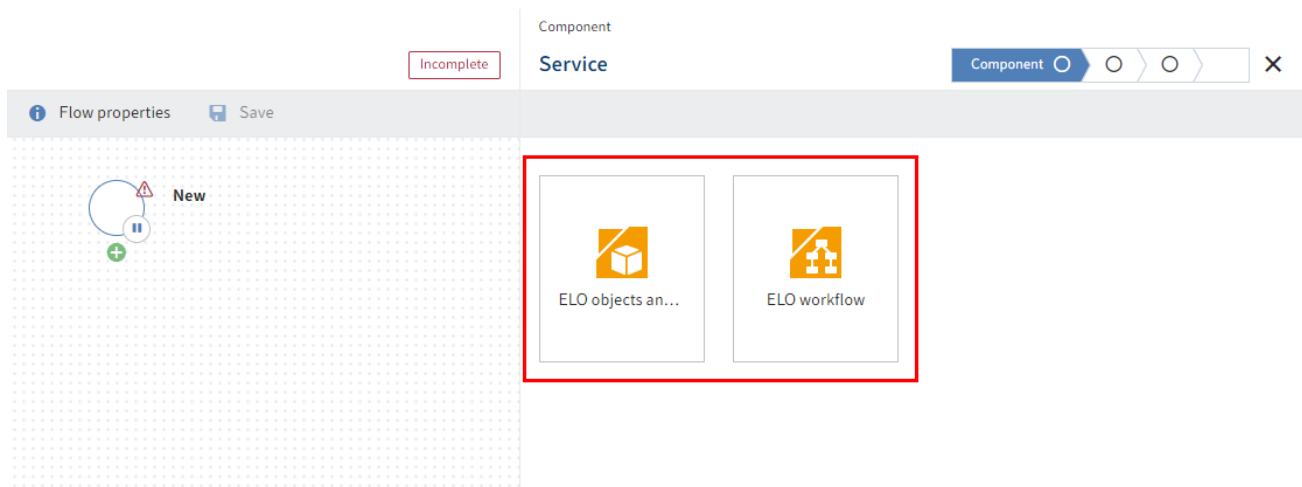
Data queries

Data queries are special flows for providing data. They run synchronously and enable you to provide data queries with the corresponding components. One application is dynamic keyword lists, for example.

If necessary, you can implement components yourself and thus provide data queries. Check the ['ELO Flows component development' documentation](#).

Select trigger

First, select a trigger. Proceed as follows:



1. Select a component whose trigger you want to use.

The flow designer automatically jumps to the following *Trigger* tab.

2. Select the trigger you want to start this flow.

The flow designer automatically jumps to the following *Settings* tab. The setting options for the trigger are shown here:

Node

trigger



Settings



ELO objects and metadata

com.elo.flows/ELOBase/v1



Query trigger with generic result



Information

The configuration data The generic configuration data Object

Input mode*

+

Key*

Value

In this example, you can change the configuration data for the *Query trigger with generic result* trigger.

Information

The section Components > ELO metadata and objects > Triggers goes into more detail on the trigger settings for the *ELO metadata and objects* component.

3. Select *Next* to open the *Summary* tab.

Node

New node

Summary X


▼ Properties

ID: 9ef81bf4-2295-419b-a506-41d7e842d7ec


Node key* 1

Node description 2

▼ Component

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1 3

▼ Service

 **Query trigger with generic result**

Endpoint: 4

▼ Settings

Node key
trigger (9ef81bf4-2295-419b-a506-41d7e842d7ec) 5

This tab shows a summary of all settings:

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

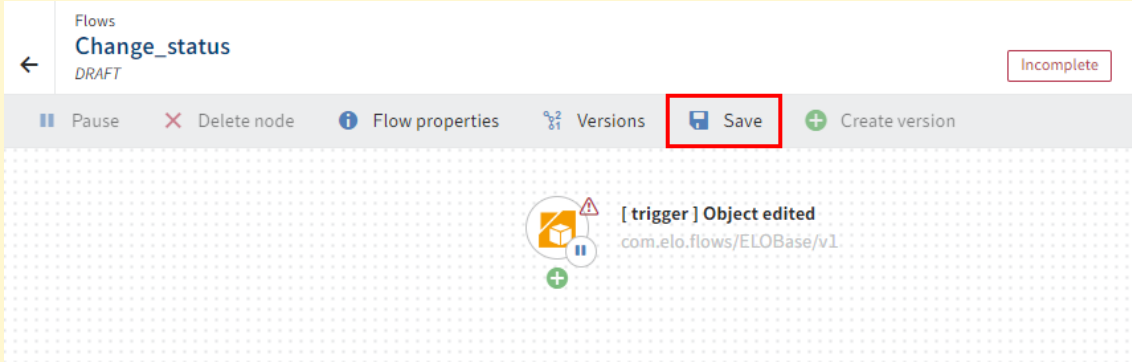
3 Component: Displays the selected component

4 Service: Displays the selected trigger with a copyable path to the endpoint

5 Settings: Displays the node key

Select *Done* to finish configuring the trigger.

Please note



With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

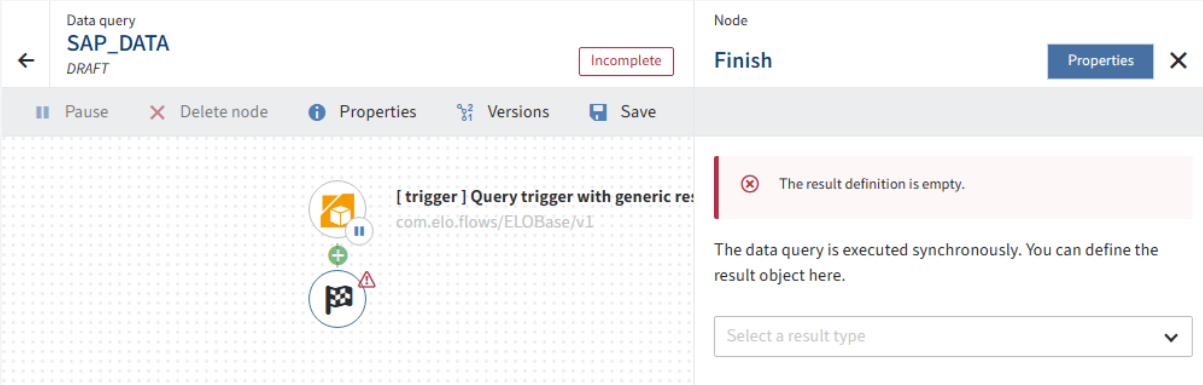
In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

5. Save the entire flow.

Information

With synchronous flows, saving changes the status of the flow to *Active*.

Synchronous flows require a result definition in the end node to switch from *Incomplete* to *Active*.



The data query is executed synchronously. You can define the result object here.

Select a result type

You will find more information about synchronous flows under Synchronous flows.

Add service

The screenshot shows the ELO Flows interface. At the top, there's a header with 'Data query SAP_DATA DRAFT' and 'Node New node'. Below the header is a toolbar with 'Pause', 'Delete node', 'Properties', 'Versions', 'Save', and 'Create versi'. The main canvas shows a flow with a trigger node '[trigger] Query trigger with generic result' and a 'New' node. A 'New node' dialog is open, showing a grid of available components: ELO objects and metadata, ELO feed, ELO workflow, FTP, and SMTP. The dialog has a red border.

Information

The close-up screenshot shows the 'New node' dialog. The trigger node '[trigger] Query trigger with generic result' is visible. A red box highlights the 'Service' and 'Branch' options in the component selection area.

Before each step, you can decide whether to create a branch with a true-false condition or to add a service. You can nest branches in other branches or continue with services.

You will find more information on branches in ELO Flows in this documentation under Advanced flow.

1. To add a service, select the plus icon next to the trigger node and select *Service*.

The configuration area with the available components opens for the new node.

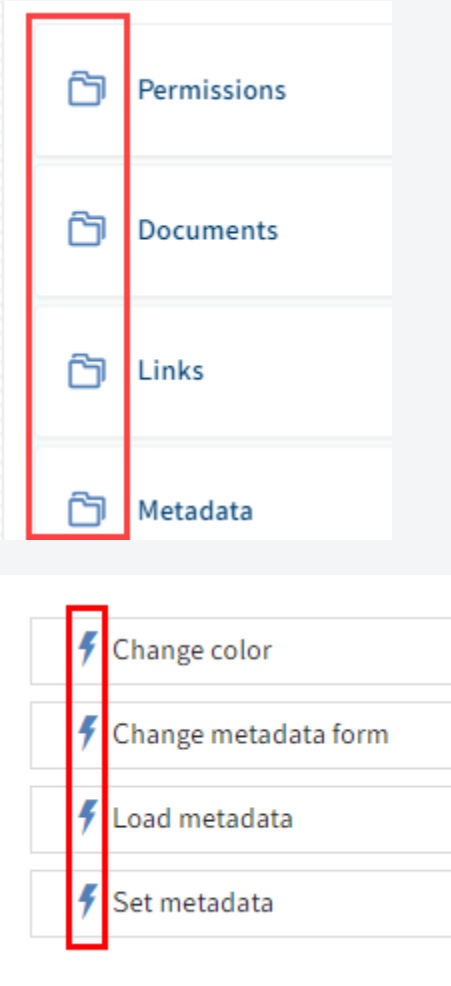
2. Select a component.

By selecting the component, you are automatically taken to the next tab with the services available for the component.

- 3.

Select the desired service.

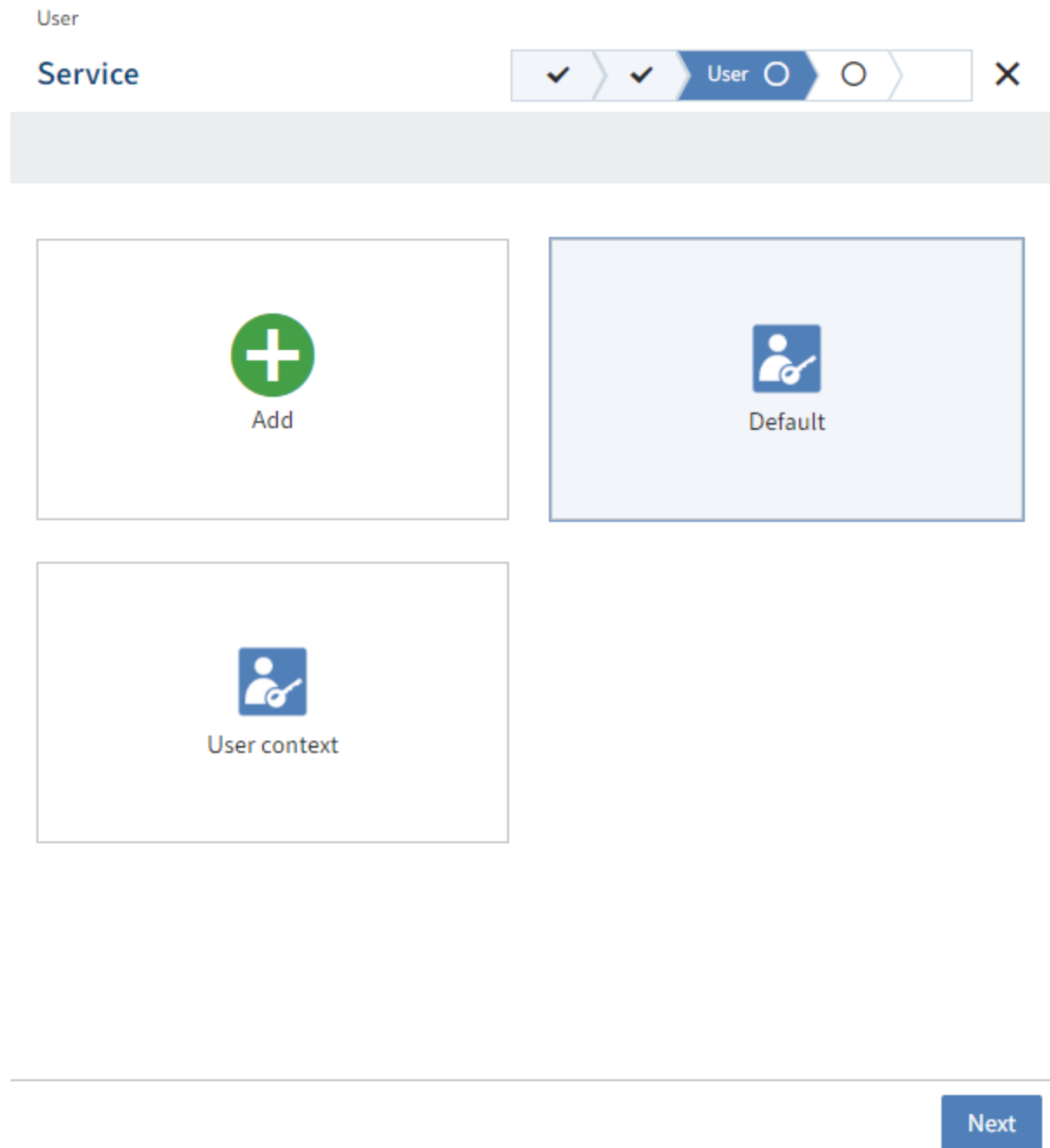
Information



Services can be listed individually (lightning icon) or in groups (folder icon).

Selecting the service automatically takes you to the next tab with the user settings.

4. Select a user connection.

**Information**

If you've already configured a user under Credentials, you can select the user here. Otherwise, you can select the default connection or add a new one.

5. Confirm your selection with *Next*.

The *Settings* tab appears.

Settings

Service



Settings



ELO feed


com.elo.flows/ELOFeed/v1



Create post



Information

Object ID* Feed post* 

Next

6. Configure custom data for the service.

Information

You can complete the fields with *static* or *dynamic values*. Dynamic evaluation is performed based on *JSONata*. With this transformation language, you can access event objects of the previous nodes and triggers in addition to using transformations or simple functions.

You will find more about the use of *JSONata* in ELO Flows under *JSONata* editor.

7. Confirm your selection with *Next*.


Information


Some services offer you the output schema function. That means that this service offers a dynamic output object. This can be the case with metadata objects, for example.

Example: You file a document with the *E-mail* metadata form and want to access a field of this document in a subsequent node. As ELO Flows initially does not know which fields are relevant for this document, the service only offers the parent object (e.g. `objKeys` and `aspectData`) in the subsequent object without access to the individual metadata fields.

Output schema

Service ✓ ✓ ✓ ✓ Output schema ○ ×

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

 Load metadata

Information

▼ **aspectData**
Metadata form

Use default Load schema


▼ **objKeys**
Metadata form


Use default Load schema


So that ELO Flows knows which metadata fields the output object has, you can either use a *standard metadata form* or load a special one. In the metadata form field, enter *E-mail* and select *Load schema*. The schema is loaded with the appropriate metadata form properties.

Output schema

Service Component ✓ Service ✓ User ✓ Settings ✓ Output schema ○ Summary ✕

 **ELO objects and metadata**
com.elo.flows/ELOBase/v1

 Load metadata

 Information

▼ objKeys

Schema

```
{
  "type": "object",
  "title": "elobasecomponent.objkeys.title",
  "description": "elobasecomponent.objkeys.descr"
}
```

Reset schema

▼ aspectData

Schema

```
{
  "type": "object",
  "title": "elobasecomponent.aspectdata.title",
  "description": "elobasecomponent.aspectdata.descr"
}
```

Next

Alternative: You can also select *Use default*. The object is then output with the default properties, and the properties of the object have to be accessed manually via JSONata.

If applicable, confirm the settings on the *Output schema* page by clicking *Next*.

Summary

Service

✓ ✓ ✓ ✓ Summary X

▼ Properties


ID 8e00ae82-ac12-4aaf-ae3a-e7cbb4e6cfe4

1 Node key* [...] step0

2 Node description

3 Do not trigger an error if the service fails.

▼ Component

 **ELO feed**
com.elo.flows/ELOFeed/v1

Done

The *Summary* tab opens. All the settings are shown once again.

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

3 Do not trigger an error if the service fails.: The entire flow runs, even if this service fails. This function is for test purposes. The result data for this service may be faulty or unavailable.

8. Select *Done* to apply the service settings in the configuration area.

Please note

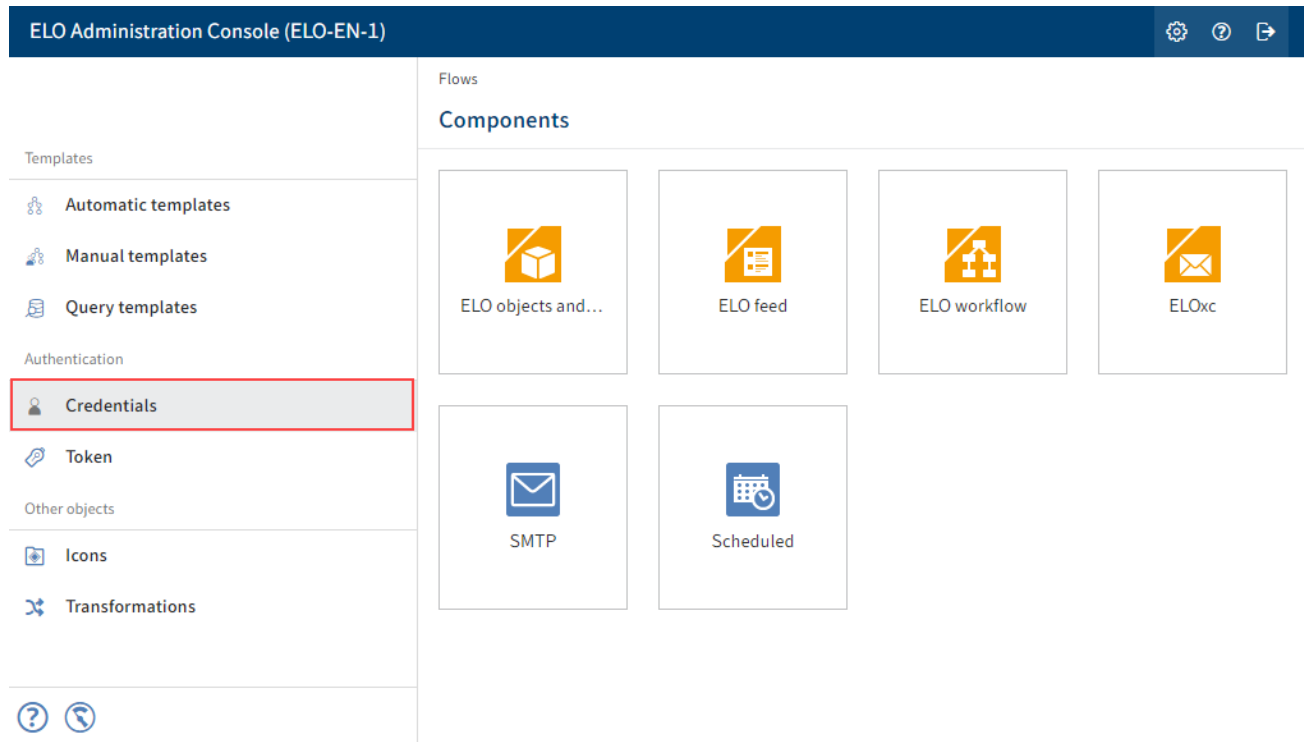
With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

9. Save the entire flow.

The status display now changes to *Active*.

Credentials



Under *Credentials*, you can configure credentials for each component.

Example applications include integrating accounts from third-party systems or configuring specific roles as users, for example invoice workflow users.

Information

Credentials are assigned to components and are not grouped into packages. They are therefore valid globally for the respective component.

1. Select a component to edit it.



The node configuration with *Credentials* opens.

You can configure the *credentials* with the following elements:

- 1 Configure new credentials
- 2 Edit a selected credential object
- 3 Delete a selected credential object
- 4 Select credential object

Information

There are components that offer the *Default* credential object. In this case, *Default* is shown as standard and assumes the credentials from the config file, e.g. the data of the ["Windows service user"](#).

You cannot edit or delete the *Default* credentials object.

Create new credentials

FTP

New credentials

Create Test

Title* Input required.

FTP connection mode FTP

Server (Host)*

Server port* Input required.

User name

Password

Anonymous logon

Home directory* /

Validate server

Information

The example image shows the credential object for the component *FTP*.

The structure of a credential object may vary depending on the component requirements. You will find more information under *Components*.

1 Create: Saves the credentials

2 Test: Tests the current entries

3 Title: Shown in the overview for the credential object

4 Configuration area: Input fields for mandatory data for the credential object

Webhooks and token authentication

You can safeguard trigger events that are triggered by third-party system calls by creating a token with an assigned user context. This check is based on the [WebSub standard](#). Third-party systems that do not implement this standard can be provided the token as a [query parameter](#) in the webhook URL.

Token administration

The token administration area provides an overview of all endpoints and their respective tokens.

1. Selecting the *Token* entry in the sidebar opens the token administration area.

All the endpoints used and their tokens are listed in the token administration area. You can also create, change, and revoke tokens here.

The endpoint entries are collapsed to begin with for a better overview of the endpoints and their respective tokens. Only when you select an endpoint does the system show a list of the tokens for exactly this endpoint.

- 1 Create: Create a new token. You will find more under [Token creation](#).

- 2 Edit: Double-click a token entry to open the editing dialog box. The user context and expiration date of the token can be changed there.

- 3 Revoke: Third-party systems that use this token are no longer authorized to trigger trigger events.

Token creation

A token requires a user context and an expiration date. This can be configured in the settings dialog box.

1. In the token administration area, select an endpoint.

The endpoint expands.

1. In the endpoint, select *Create*.

The *Token* dialog box opens.

The screenshot shows a dialog box titled "Token" with a close button (X) in the top right corner. Below the title bar is a "Create" button. The dialog contains three input fields:

- Resource:** A text field containing "com.elo.flows/ELOxc/v1/XcActionExecuted" with a "Copy" button to its right.
- Valid until*:** A date field containing "04/30/2023" with a green checkmark to its right.
- User context*:** A dropdown menu showing "User selection" with a downward arrow.

1. In the *Valid until* field, enter how long the token should be valid.
2. Select *Select users*.
 - A search field opens.
3. Search for the user you want the token to run through.



Information

The user must be created in the *Credentials* menu item.

4. Select a user.
5. Select *Create*.

ELO Flows creates a new token.

Token ✕

 Save  Revoke

Resource Copy

Valid until* ✓

User context* ▼

Secret Copy

Make sure you copy the secret. It will never be shown again.

After creation, the secret used for webhook registration is shown in the dialog box.

Important

The secret cannot be viewed again. Create a copy for this reason.

Query parameter

Not every third-party system supports the WebSub standard. To be able to use this kind of system anyway, you can provide a query parameter containing the token secret to the webhook URL. This URL is then used to register the webhook in the third-party system. As a result, the secret is provided from the third-party system when the trigger event is triggered for authentication.

A possible endpoint can look like this, for example:

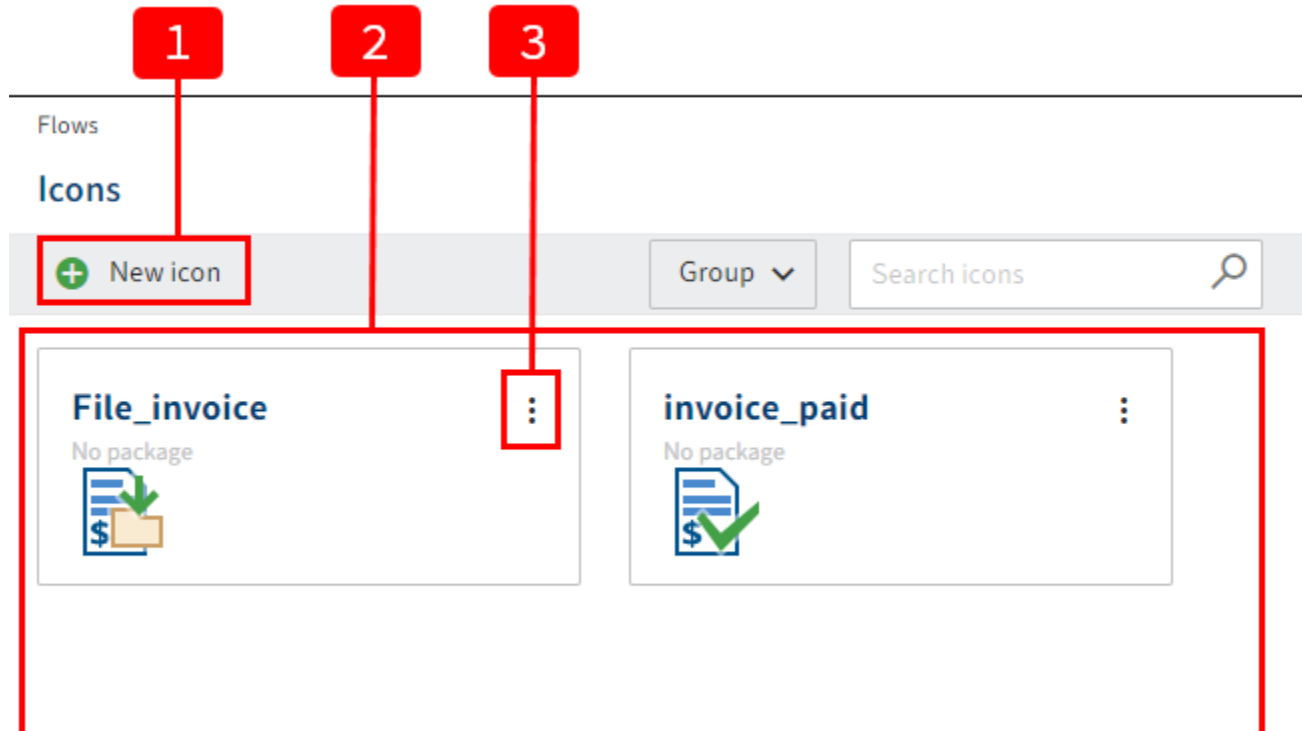
```
http://<server>:<port>/ix-<repository>/plugin/de.elo.ix.plugin.proxy/  
flows/api/v1/trigger/com.elo.flows/EL0Base/v1/ObjectCreated?token=07Cz3CCKYiW320XrRuJZ0bFjyou
```

Please note

Query parameters should only be used in the event of a possible HTTPS connection.

Icons

Select *Icons* to see all the icons available for ELO Flows. These icons can be used for *user actions* for buttons in the ELO clients.



You have the following configuration options:

- 1 Add new icon
- 2 Select existing icons
- 3 Edit or delete an existing icon

Information

You cannot change the name. Create a new entry if you need an icon with another name.


Create icon


1. Select *New* to create a new icon.

Leave_request

Create icon ✕

- 1
- 2 Identifier* ✓
- 3 Package ▾
- 4 Icon* ⓘ



Drop file here or click to upload.
- 5 Preview 

You have the following configuration options:

1 Create: Saves the configuration

2 Name: Unique, technical name of the icon

Please note

The name cannot be changed later.

Information

Do not use special characters.

The following characters are allowed:

- a-z
- A-Z
- 0-9
- äöü
- ÄÖÜ
- ß
- _

3 Select package: Assigns the icon to a package as needed

4 Icon: Upload image file

Information

- Maximum size: 1 MB
- Format: SVG

Transformations

Transformations are used to assign single or multiple input parameters for ELO Flows services. These assignments are based on input objects defined in the ELO Flows components. Transformations consist of a source object, a target object, and their mapping.

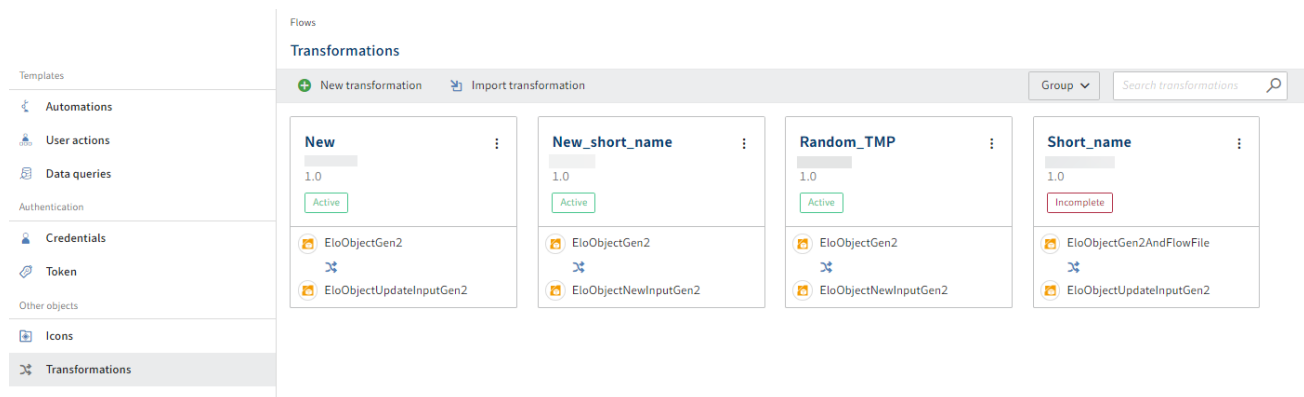
One use for transformations is to save complex assignments and to reuse them. For example, you can transform data objects with different data models onto the aspect data model of ELO objects or, in another context, transform the data model of ELO objects onto other target data models.

In addition, you can use transformations to change assignments across multiple flows at once.

To use transformations, create global ones in the [Transformations](#) area.

Alternative: You can also create transformations [in the configuration area](#) while creating a flow.

Create global transformation



1. Open the *Transformations* area.
2. Select *New transformation*.

The *Create transformation* dialog box opens.

3. Enter a unique name for the transformation.

Information

Do not use special characters.

The following characters are allowed:

- a-z
- A-Z
- 0-9
- äöü
- ÄÖÜ
-

β

○

—

The transformation is automatically assigned to the package you are currently configuring.

4. Select *Create*.

The *Transformation* dialog box opens.

Information

The respective drop-down menu for the source object or target object is based on the components you can use to perform a transformation.

5. In the *Source* field, select the source object that you want to link.
6. In the *Target* field, select the target object you want to transform.
7. Adapt the transformation object fields.

The configurable fields are made available based on the target object.

8. Select *Save transformation*.

Create a transformation in the component context

You can create and edit transformations while designing new flows.

Node

step1

Component ✓

Service ✓

User ✓

Settings ✓

Output schema ✓

Summary

✕

File*

activities.file.file

J:



Date, extra text, form, short, name and owner name are completed via transformation.



Apply transformation

Context*

step0



Transformation*

Short name



Apply transformation

Type of entry ⓘ

J:



Color ID ⓘ

J:



Information

You will recognize whether a component allows for transformations based on the *Apply transformation* switch.

1. In the configuration area, select *Apply transformation*
2. In the *Context* field, select the source object for the transformation.
3. Select the plus icon to create a new transformation.

Alternative: Select a global transformation configured for the source object.

Information

You can only apply transformations if the source and target object are configured in the component code as *transformable*.

4. In the *Transformation* dialog box, enter a unique name for the transformation and enter the values you want to assign.

Flows

TRANS
DRAFT

←

Pause

Transformation

×

Create transformation Cancel

Identifier* 0 / 128

Package

Source* EloObjectGen2 com.elo.model (EL... ▼

Target* EloObjectNewInputGen2 com.elo... ▼

▼ Date

Date ⓘ j | Az | ↵

Information

The *Source* and *Target* fields are set automatically. The contents there are based on the context entered previously and the service just configured that you want to apply the transformation in.

5. Select *Create transformation* to save your entries.

The newly configured transformation is assigned to the flow and also created as a global transformation in the *Transformations* area.

JSONata editor

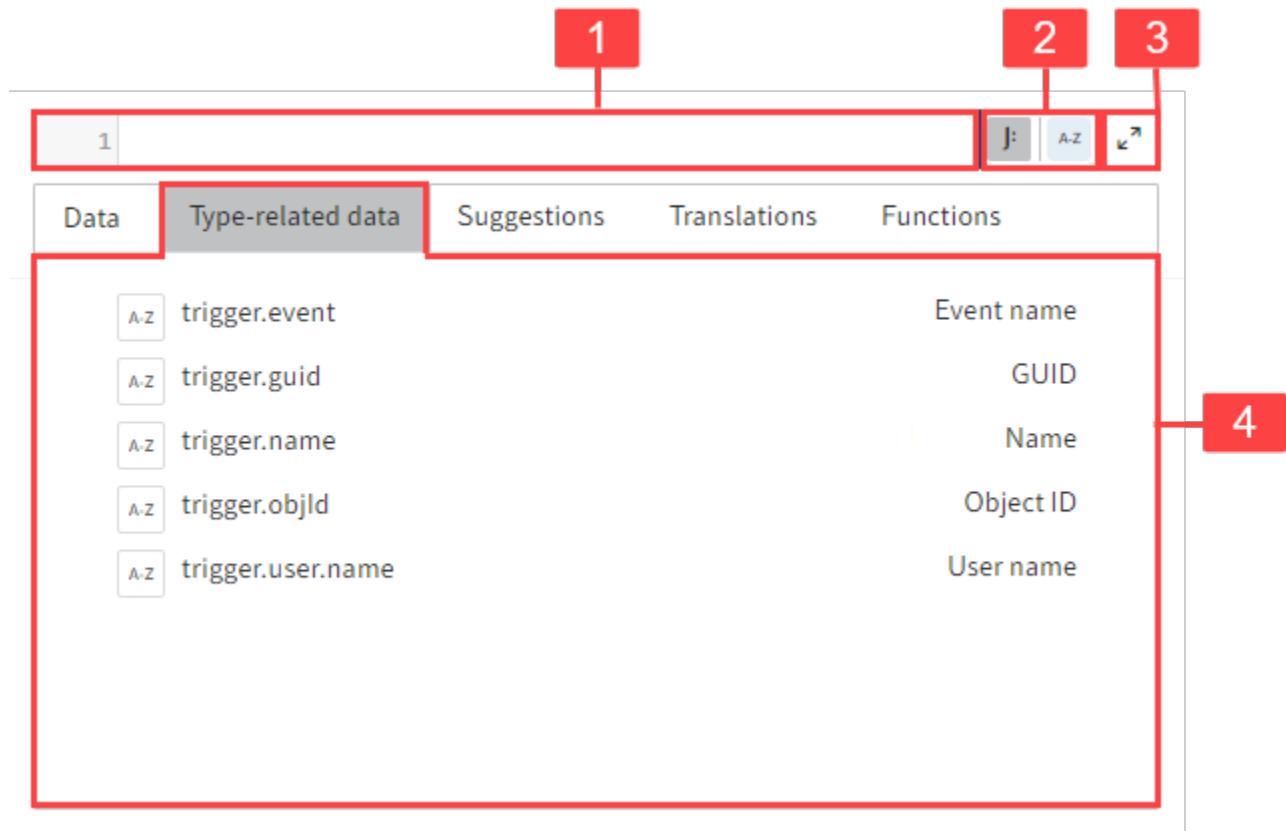
With the *JSONata editor*, define the *Service properties* of the flows. If you create a service, in most cases the service will need data it can use to execute the respective action.

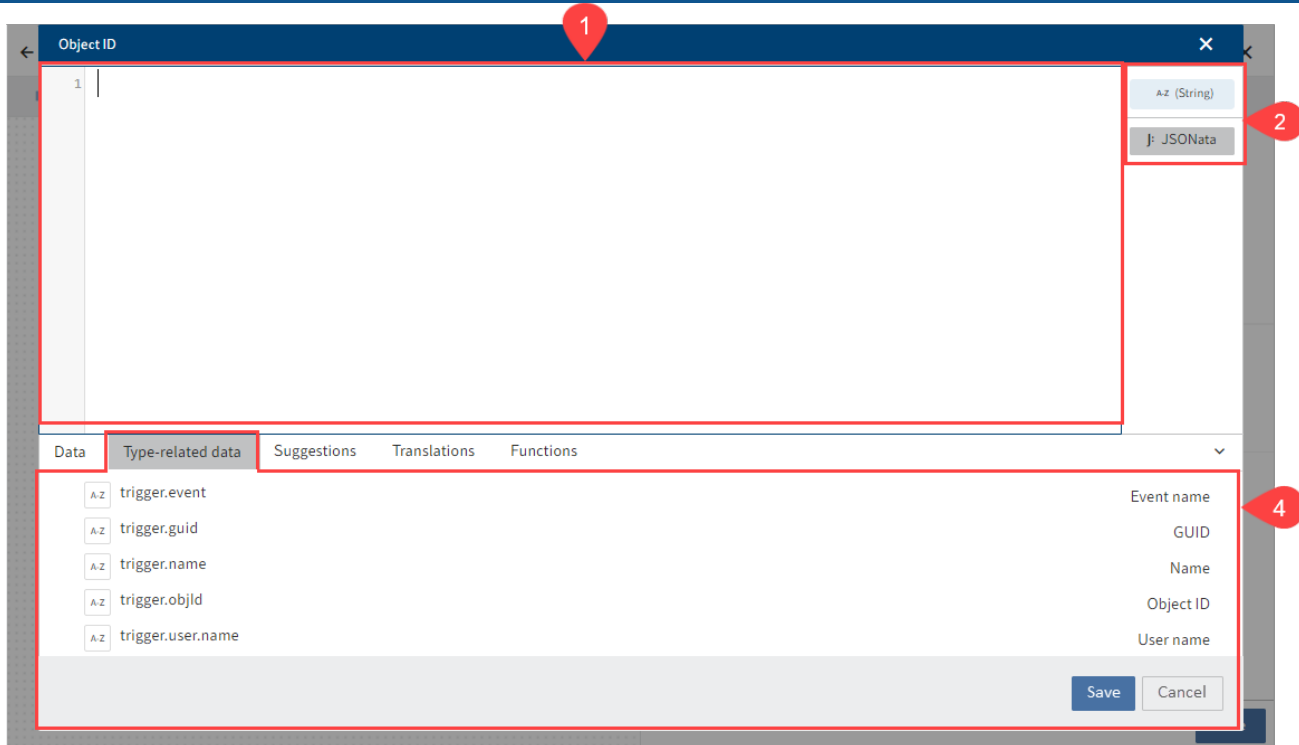
Example

One example is the object ID of an ELO object. This is used in the *Create post* service of the ELO feed component:

The service needs to know which object a feed post should be written for. The object is addressed via the object ID, which you can define or identify dynamically.

ELO Flows uses the transformation language JSONata for dynamic data access with data transformation.





You have the following configuration options:

1 Enter value: Input field for fixed values as a string or dynamic values as a JSONata expression.

2 Select input mode: The input field is in text mode by default. Text mode works with string values, which you can select from configured data (4). Expert mode (J icon) enables you to input JSONata commands.

3 Expand editor: Expand the editor using the arrow icons and open a separate dialog box for a larger input area.

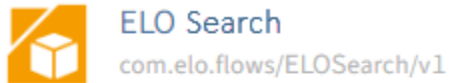
4 Use preconfigured contents: Choose between *Data*, *Type-related data*, *Proposals*, *Translations*, and *Functions*.

Information

For more information about the structure and possibilities with JSONata, refer to jsonata.org

Advanced flow

Multi-instance trigger



i Information

⚙️ Perform metadata from search

🔍 Multi-instance

i

Use multi-instance triggers to process flow instances in stacks. One flow instance is started for each object returned by the trigger.

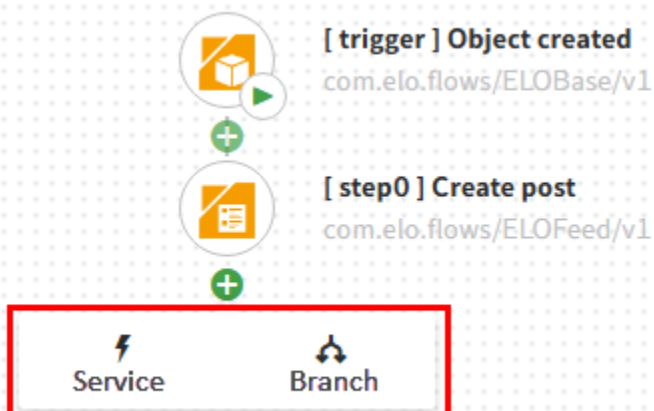
Branches

In ELO Flows, branches enable you to generate more complex workflows with a flow template via a true/false query. At the branches, ELO Flows checks a binary condition and runs the relevant flow based on the feedback.

Configure branches

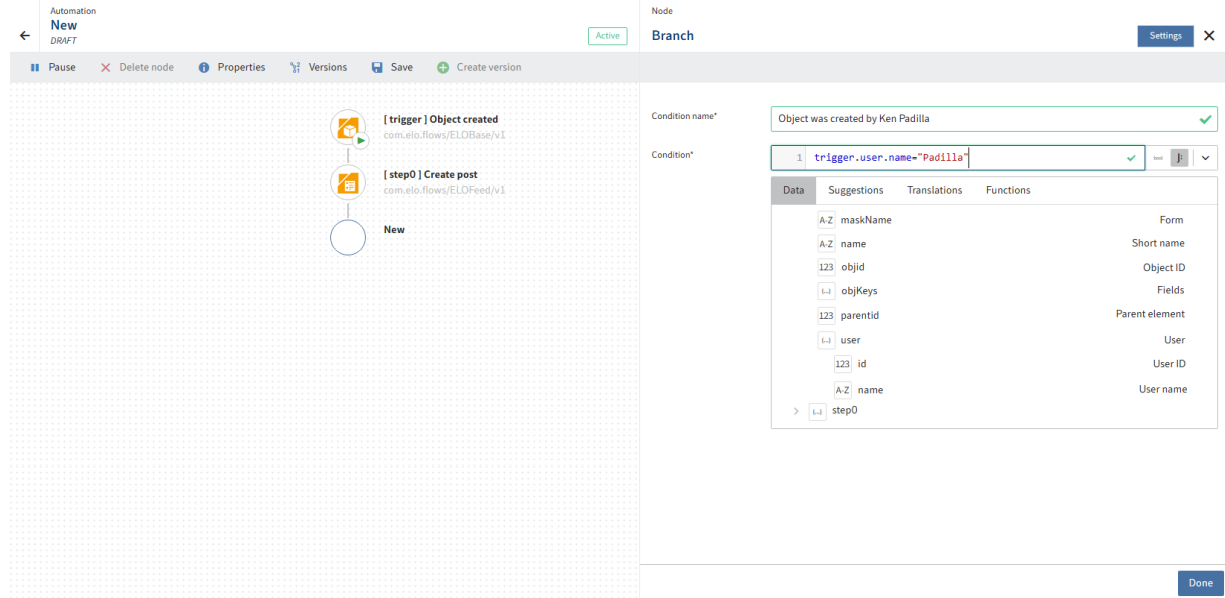
When creating your flow, decide whether you want to create a branch or a service after each node.

Method



1.

Select *Add node* (button with plus icon) and select *Branch*.



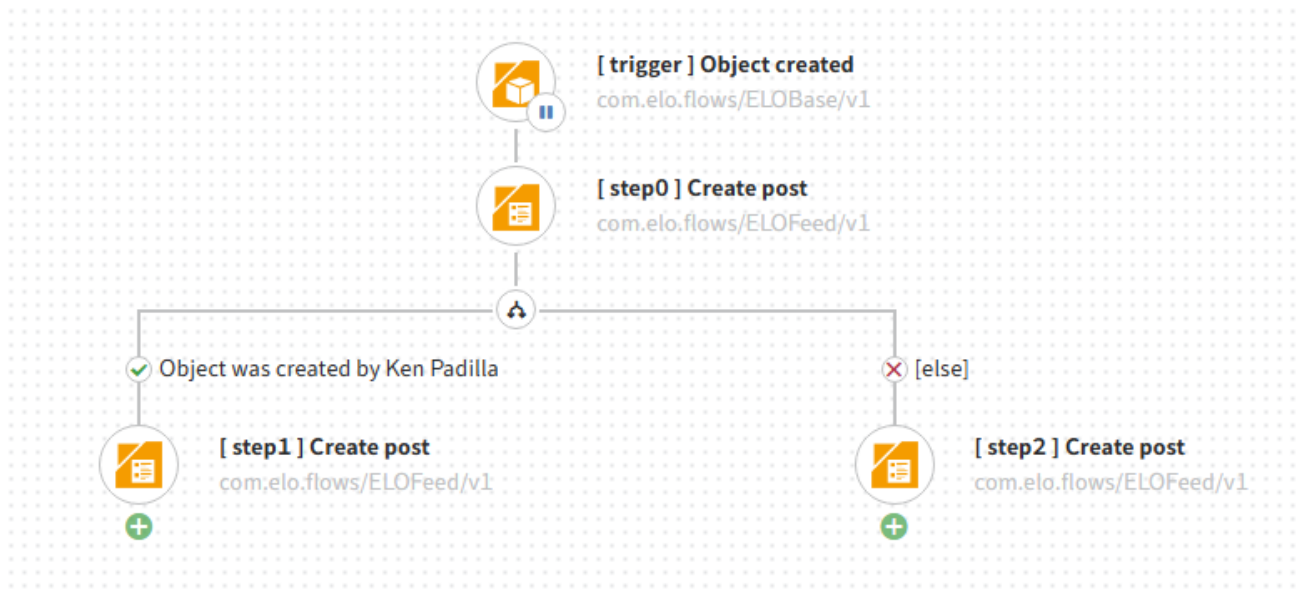
The screenshot shows the 'Node' configuration panel for a 'Branch' node. The 'Condition name' is 'Object was created by Ken Padilla'. The 'Condition' is 'trigger.user.name='Padilla''. Below the condition field, there is a table with columns 'Data', 'Suggestions', 'Translations', and 'Functions'.

Data	Suggestions	Translations	Functions
A-Z	maskName		Form
A-Z	name		Short name
123	objid		Object ID
[...]	objKeys		Fields
123	parentid		Parent element
[...]	user		User
123	id		User ID
A-Z	name		User name
>	step0		

The node configuration for the new *branch* opens.

2. Enter a title for the branch condition. After you confirm the change in the node configuration, this title appears for the true condition in the flow designer.
3. In the JSONata editor, configure the *condition* that the binary branch refers to. These conditions determine which branch is run over the course of the flow.
4. Confirm with *Done*.

Result



Information

You can create the branch node at the end of the existing flow or between two existing nodes.

In a synchronous flow, an end node is automatically placed at the end of each branch when creating a branch.

Refer to the Synchronous flows section in this documentation for more information.

The node configuration closes and you are back in the flow designer.

The branch with the true condition now has the name of the title entered for the branch condition in the flow designer.

You can continue configuring your flow.

Select *Add node* to create either additional branches or services at the end of the branch.

Information

You do not have to create nodes at both ends of the branch. If an empty branch is run at runtime, the flow ends at this point. If both branches are empty, the binary branch is ignored.

Delete branches

The screenshot displays the flow designer interface. At the top, there is a navigation bar with 'Automation', 'New DRAFT', and 'Node Branch' tabs. Below this is a toolbar with buttons for 'Pause', 'Delete node' (highlighted with a red box), 'Properties', 'Versions', 'Save', and 'Create version'. The main workspace shows a flow diagram with three nodes: a trigger node '[trigger] Object created', a step node '[step0] Create post', and a branch node. The branch node has two outgoing paths, one labeled 'Object was cre...' and another labeled '[else]'. To the right of the flow diagram is a configuration panel for the selected branch node, showing 'Condition name*' as 'Object was created by Ken Padilla' and 'Condition*' as 'trigger.user.name="Padilla"'. There are also 'test' and 'j' buttons in the configuration panel.

As with all nodes in the flow designer, you can configure or delete a selected branch node after it has been created.

Selecting *Delete node* for a branch node opens a dialog box. Here, you can choose whether you want to delete single branches (1, 2) or both branches (3). In case 3, all steps following the branch are deleted.

Delete node

**Delete branch node**

Choose one of the following options to delete a branch node:

**Delete True branch (left)**

The True branch (left) is deleted and the False branch (right) is maintained.

1

**Delete False branch (right)**

The False branch (right) is deleted and the True branch (left) is maintained.

2

**Delete both branches**

Both branches are deleted.

3

Cancel

Information

When creating a flow with branches, there are three possible situations that can have an effect on how the branch node is deleted.

Both ends of the branch are empty: If both ends of the branch are empty, the entire branch node is deleted. The flow ends with the last configured node before the branch was inserted.

One end of the branch is empty: If only one end of the branch contains more nodes, the branch is inserted in the flow in place of the branch node.

There are configured nodes at both ends of the branch: If both ends of the branch have configured nodes, you can decide whether to delete the entire branch node or a specific end of the branch. If you delete the entire branch node, the nodes that follow the branch are also deleted. If you delete one branch, the other branch is inserted in the flow in place of the branch node.

Synchronous flows

Flows configured as synchronous flows are executed with feedback. This enables you to show an info box in the ELO client once a flow has been completed, for example.

All flows are asynchronous by default. This means that the flows are executed without final feedback. This may make sense if you want to prevent the user interface from being blocked, for example.

The following section explains how you can enable synchronous feedback for user actions and data queries and recognize whether synchronous feedback is supported.

User actions with synchronous feedback

For user actions, you enable synchronous feedback in the node configuration for the manual trigger.

Node

Manual trigger



Summary



▼ Properties

ID

3f96da0b-86c4-4edb-9ae8-8790afca9320

Node key*

{...} trigger

Node description

 Enable synchronous feedback

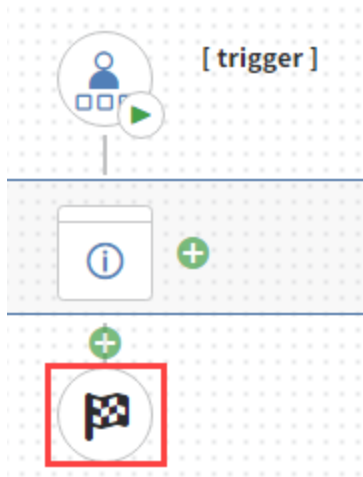
> Action definition

> Service

> Settings

Done

1. In the configuration for the manual trigger, check the *Enable synchronous feedback* check box in the *Summary* step.



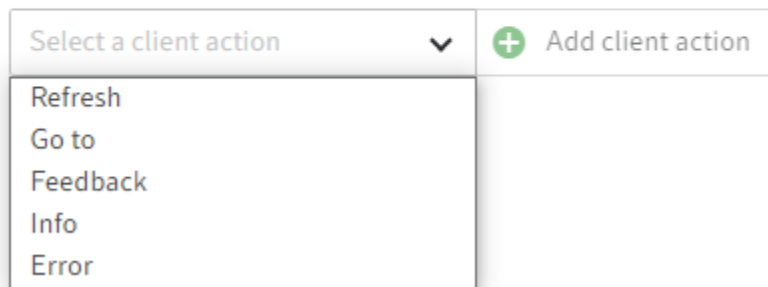
2. Finish your configuration.

An end node (button with flag icon) appears in the flow designer.

3. Open the node configuration for the end node.

Enable synchronous feedback

You must add at least one client action for synchronous feedback to remain enabled.



You can add different client actions via a drop-down menu here:



- Refresh: Refreshes the ELO client once the flow is processed.
- Go to: The ELO client goes to the configured object.
- Feedback: The ELO client shows a toast with the configured message.
- Info: The ELO client shows an info box with the configured message.
- Error: The ELO client shows an error box with the configured message.

4. Select an action.

5. Select *Add client action*.

Enable synchronous feedback

Select a client action ▼ + Add client action

 Feedback


Shows a feedback message in the client.

Object properties

Text*

Information

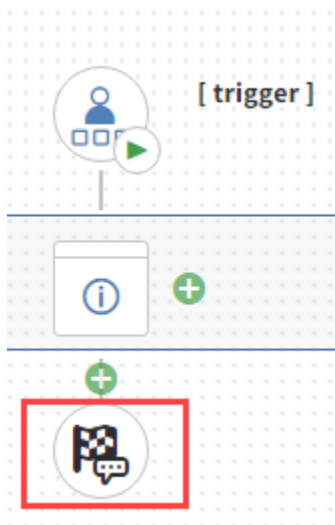
You can link multiple client actions.

However, not all combinations make sense.

Example: A *Go to* action implies a *Refresh* action. So, you don't have to link them.

6. Configure the desired object properties.

7. Select *Done*.



The selected action(s) are linked with the flow. The flag icon changes, indicating that client actions have been configured.

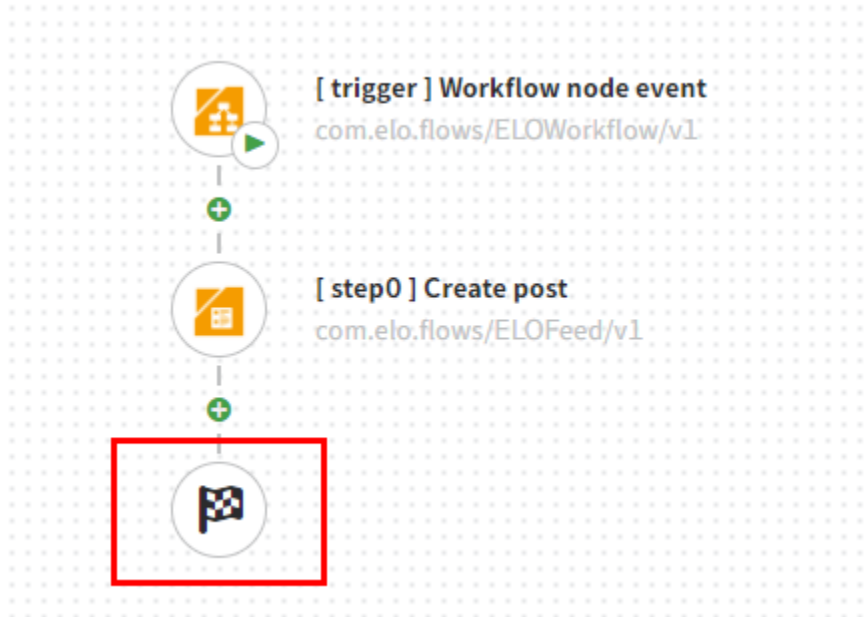
8.

Save the flow.

The status display changes to Active.

Automations with synchronous feedback

With automations, there are triggers that support synchronous feedback and triggers that don't.



The flag icon automatically appears in the flow designer for triggers that support synchronous feedback.

Properties

Finish

Properties



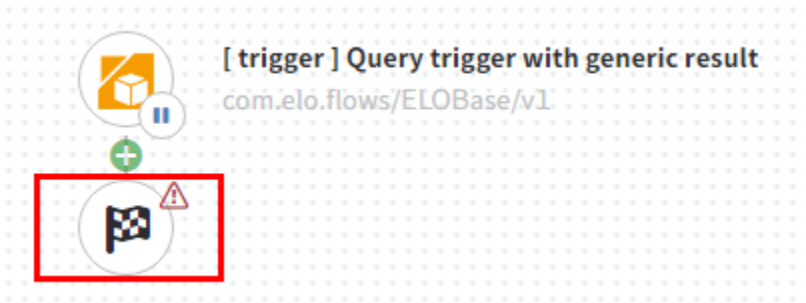
The flow is executed synchronously. The trigger does not define a result but returns a standard result.

Information

Unlike with *user actions*, no properties can be configured for synchronous feedback in *automations*. In this case, the trigger defines the result. The flag symbol here is only intended as an indicator that there is a configuration.

Data queries with synchronous feedback

Data queries with synchronous flows require a result definition.



1. Select the end node (flag symbol).



The data query is executed synchronously. You can define the result object here.

GenericQueryResult ▼

Object properties

Data The generic query result Object

Input mode* Positions ▼

+

Key* user

Value cole ab J: ▼

- 2.

In the node configuration for the end node, select a result type from the drop-down menu.

Optional: Edit the object properties if necessary.

3. Select *Done* to apply the result definition.

The selected result definition is linked with the flow. The flag icon changes, indicating that there is a configuration.

4. Save the flow.

The status display changes to Active.

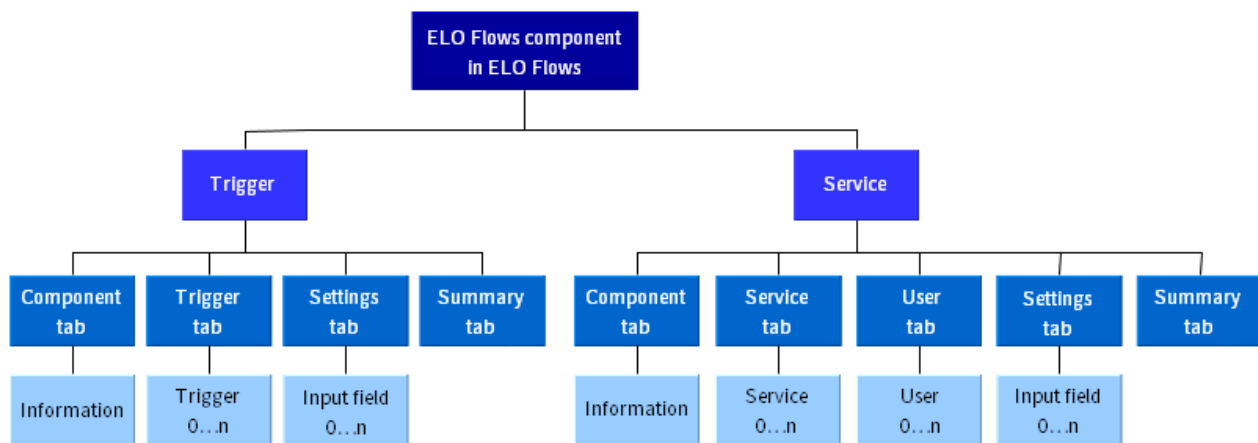
Components

What are components in the context of flows?

In flows, components are self-contained software elements that extend ELO's standard functions.

The components can be programmed independent of customer systems and integrated into the respective customer system as needed via standardized interfaces.

In the context of flows, a component is output as a JAR file and provided for further use in the ELO Flows worker.



Components can be triggers, services, or a combination of the two. In addition, information fields can be integrated into flows.

ELO provides several components as standard. You can also develop custom solutions.

The following describes the components provided by ELO as standard.

Install new or custom components

To provide custom components in the flow administration area, roll out the JAR file in the flow worker (Apache Karaf). It can either be deployed via the *deploy* directory of the ELO Flows worker or, in the cloud context especially, via the repository in the ELO client.

Deployment via the *deploy* directory of the flow worker

Copy the JAR file of the new component to the *deploy* directory. A possible path to this directory could be: ELO\servers\ELO-Flows-Worker-1\deploy.

Deployment via drag-and-drop in the ELO client

- 1.

Drag the JAR file of the component to the repository and drop it in the *Administration//ELOflow Base//CustomComponents* area.

2. Assign the JAR file the *ELOScripts* metadata form in the ELO client.

For more information on development, refer to the [ELO Flows component development](#) documentation.

General information

Input fields in the configuration area

Components are configured using input fields among other things. The following goes into more detail about the input variants.

Determine the ARCPATH syntax and object ID

With an ARCPATH, you can specify a start or target element via a folder path. The path can also be specified relative to a SORD.

Type	Description	Example
<ID> <GUID>	Enter a static ID or GUID.	Static ID: 1234 GUID: (E10E1000-E100-E100-E100-E10E10E10E00)
ARCPATH: <path>	After the colon, enter the filing path. The first (any) character always separates the level in the folder structure.	ARCPATH:¶Accounting¶New
ARCPATH[<GUID>]:<path> ARCPATH[<ID>]:<path>	Enter an object ID or GUID in square brackets before the colon. A path is then identified relative to the start point.	ARCPATH[(E10E1000-E100-E100-E100-E10E10E10E00)]:¶Business Solutions Custom¶Configuration
OKEY:<field name>=<value>	Searches for an exact entry in a field. The value should be unique. Only an object ID is returned.	OKEY:ELOINDEX=incinvoice
LMATCH:<field name>=<fixed value>%<search string>	Similar to OKEY. Searches for the exact fixed value and longest matching search string in the named field.	LMATCH:CUSTOMER_NAME=Contelo%Holdings

Using gen. 1 and gen. 2 metadata

With flows, you can use and process gen. 1 and gen. 2 ELO-specific metadata.

Type	Metadata gen.	Description	Example
Metadata form	(gen. 1)	Identify the metadata form with the name of the metadata form: <MASK_NAME>	INVOICE
	(Gen. 2)	Identify the metadata form with the name of the package. Enter the name of the package as the prefix, separated from the name of the metadata form with a period: <PACKAGE_NAME>.<MASK_NAME>	BASIC.PERSON
Data field	Index field (gen. 1)	Set the index field via the field group name (<i>Key</i>) <FIELDGROUP_NAME> and the field value (<i>Value</i>) <FIELD_VALUE> In ELO Flows, index fields are also referred to as <i>ObjKey fields</i> .	Key: INVOICE_NO Value: 42
	Aspect field (gen. 2)	Identify the aspect field with the name of the aspect: <ASPECT_NAME>.<ASPECT_FIELD_NAME>	PERSONAL_DATA.FIRSTNAME

'Input mode' in the context of lists and objects

The screenshot shows a configuration window titled "Attachment" with a green "Array" button in the top right corner. The main area contains a label "Input mode*" followed by a dropdown menu. The dropdown menu is open, displaying four options: "Positions", "Basic entry", "Positions", and "List transformation". The second "Positions" option is highlighted in blue. Below the dropdown menu is a text input field labeled "file 0" with a search icon and a dropdown arrow.

Generate new lists and objects in the context of flows.

ELO Flows offer you *Input modes* in the configuration area to support you when designing lists or objects.

When selecting an input mode, you can decide on the support you want when you are making the entry. The input modes *Positions*, *List transformation*, and *Basic entry* are available.

Positions

The screenshot shows the 'Files' configuration panel. At the top right, there is a green 'Array' button and a small icon. Below this, the 'Input mode*' dropdown is set to 'Positions'. A large grey button with a plus sign is centered below the dropdown. Underneath, there are two input fields labeled 'file 0' and 'file 1'. The 'file 0' field contains the text 'activities.file.file' and has a JSONata editor icon (J: [] ↵) to its right. The 'file 1' field is empty and also has a JSONata editor icon to its right.

Create a list or an object with a fixed number of entries manually. You can fill the entries statically or dynamically. You determine the size of the list or object statically.

Add the desired number of entries using the plus button. Use the JSONata editor in the input fields to integrate dynamic field values.

You can find more information about the JSONata editor in flows in the *JSONata editor* section.

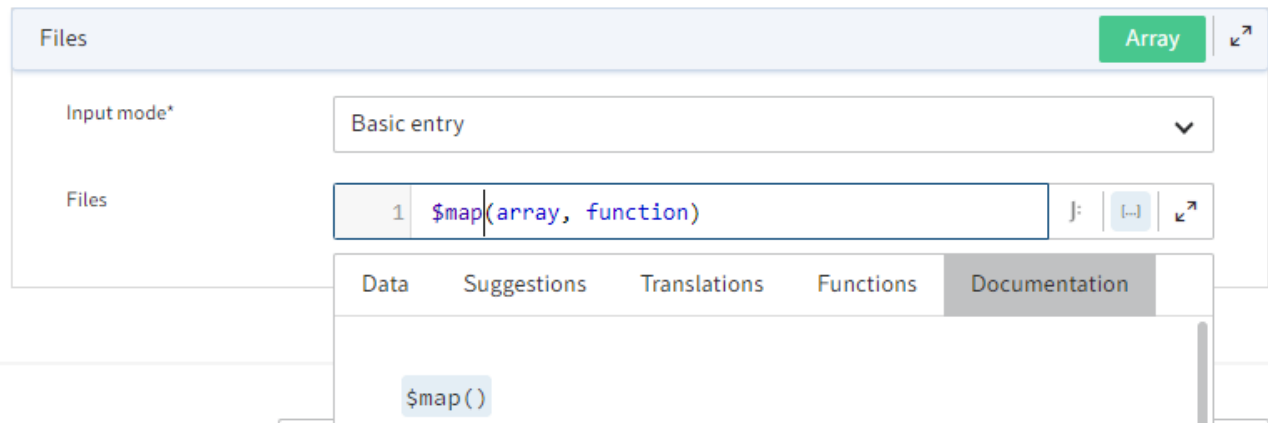
List transformation

The screenshot shows the 'Files' configuration panel. At the top right, there is a green 'Array' button and a small icon. Below this, the 'Input mode*' dropdown is set to 'List transformation'. The 'Context*' field (with an information icon) contains the number '1' and the text 'activities.files.files'. To the right of the context field is a JSONata editor icon (J: [-] ↵). Below the context field, there is an input field labeled 'file' which is empty, and to its right is a JSONata editor icon (J: [] ↵).

Design a new list dynamically on the basis of an initial list.

Enter the initial list in the *Context* field and in the next field, define how the elements will be transformed in the initial list.

Basic entry



Generate a new list or a new object entirely using JSONata commands. JSONata allows you to use both static and dynamic processes.

You can find more information about the JSONata editor in ELO Flows in the *JSONata editor* section.

ELO objects and metadata



Goal and use

The *ELO objects and metadata* component provides you with basic functions for communicating with an ELO repository.

Credentials and connection data

Enter connection data for the *ELO objects and metadata* component. A *default connection* is available. This is entered in the config files. The *user context* assumes the session of the user logged into the ELO client. You can also create additional connections to run services with defined rights.

Triggers

Triggers are the starting points of a flow. The *ELO objects and metadata* component offers the following triggers:


Name	Event	Additional information
Query trigger with generic result	Started when the endpoint is called with an HTTP POST. The trigger can be started with gen. 1 and gen. 2 metadata. Metadata (gen. 1):	Synchronous flow
Dynamic keyword list called	Configure dynamic keyword lists with a flow call in the <i>Form designer</i> or in the <i>Field templates</i> . Metadata (gen. 2): Configure dynamic keyword lists with a flow call in the aspects.	Synchronous flow
Object created	Started when an object is filed in ELO.	Asynchronous flow
Object edited	Started when an object in ELO is changed (e.g. metadata or versions).	Asynchronous flow


Trigger configuration 'Query trigger with generic result'


Node

trigger

Component ✓ Trigger ✓ Settings ○ Summary ✕

 ELO objects and metadata
com.elo.flows/ELOBase/v1

 Query trigger with generic result

 Information


The configuration data Object

You can transfer configuration data via a REST call or forego the transfer of configuration data entirely. To do this, do not make any additional configurations on the *Settings* tab.

Advanced settings

The configuration data The generic configuration data Object

Input mode* Positions



1 Key*

2 Value

Optional: Enter configuration data via the *Positions* input mode, e.g. data from a third-party system. Enter your user-defined data in the key-value list. The key (1) is a mandatory field, while the value (2) is optional.

Example entry

Key - user

Value - renz

Next steps with the 'Query trigger with generic result' trigger


This trigger is started when the endpoint is called with an HTTP POST. You can get the endpoint under *Summary > Service*.


'Dynamic keyword list called' trigger configuration


Node

trigger

Component ✓ Trigger ✓ Settings ○ Summary ✕

 ELO objects and metadata
com.elo.flows/ELOBase/v1

 Dynamic keyword list called

 Information

Input data types Array

Call dynamic keyword lists with a flow call. The trigger can start in the context of gen. 1 and gen. 2 metadata. The call is structured as follows: flows-plugin/trigger/trigger ID and may look like this, for example: flows-plugin/trigger/08e40e34-2e3e-4725-baed-1442d1e25143.

For gen. 1 metadata, the data entered is not automatically transferred formatted. You can configure formatting on the *Settings* tab as needed. Gen. 2 metadata automatically provide formatting.

Configure flow call for dynamic keyword lists

To call dynamic keyword lists with flows, the flow call has to be configured in the metadata. The following explains more for both gen. 1 and gen. 2 metadata.

Metadata (gen. 1):

The trigger is started when an ELO Flows call is used to store dynamic keyword lists in the *Form designer (gen. 1)* or in the *Field templates (gen. 1)*.

You can read the trigger ID in the flow designer for the trigger under *Summary > Properties > ID*.

Configure in the Form designer:

1. Select a field.
- 2.

Open the properties for the selected field.

- Under **Keyword list**, select the option *Dynamic Keyword Map*.
- In the *Script name* input field, enter the call according to the convention `flows-plugin/trigger/Trigger-ID`.

Store in the Field templates:

- Select a field.
- Enter the call under *Keyword list > Dynamic keyword list*.

Metadata (gen. 2):

The trigger is started when a flow call is used to store dynamic keyword lists in an aspect field.

- Within your package, navigate to *Aspects*.
- Select an aspect or add a new one.
- Within the aspect, navigate to the *Fields* area.
- Select a field or add a new one. The settings for the field open.
- In the *Dynamic keyword list* input field, enter the call according to the convention `flows-plugin/trigger/Trigger-ID`.

Advanced settings

The screenshot shows a configuration window titled "Input data types" with a subtitle "The input data types for gen. 1; if no configuration is made for a form field, this field is not normalized." and an "Array" button in the top right. A plus sign is centered in the main area. Two red boxes with numbers 1 and 2 point to the "Entry type*" and "Field name*" fields respectively. The "Entry type*" dropdown menu is open, showing options: Integer, Number, Date, and Datetime.

Optional: Manual formatting of gen. 1 metadata entries

1 Entry type: Select an entry type from the drop-down menu. The following options are available: Integer, Number, Date, Datetime.

2 Field name: Enter the field name with dynamic keyword list from the context of forms with gen. 1 metadata.

Example entry

Entry type - Date

Field name - Filing date

Next steps for 'Dynamic keyword list called' trigger

Use the result *SimpleDynKwlColumnsResult* or *DynKwlColumnsResult* at the end of the flow under *synchronous feedback*.

Result definition via 'SimpleDynKwlColumnsResult'

Node

Finish Properties ✕

The automation is executed synchronously. You can define the result object here.

SimpleDynKwlColumnsResult

Object properties

1 Title of table* ⓘ

2 Columns The title of the table Array

3 Column title* ⓘ

4 Form field name* ⓘ

5 Column type* ⓘ TEXT

6 Column data The column data Array

7 Context* ⓘ

8 string

Configure the result object manually with the following object properties:

- 1 Title of table: The *Title of table* field is mandatory. Enter the title of the generated results table.
- 2 Columns: Open the advanced configuration for the columns by selecting the plus icon.

Information

Manual configuration of individual columns of the result table is optional. If you decide to configure columns, the fields *Column title*, *Form field name*, and *Column type* are mandatory.

3 Column title: Enter a title to be displayed for the column. If you don't want to show a column title in the table, enter null.

4 Form field name: Enter the name of the form field the data should be transferred to. If you don't want to transfer the data from this column to a form field, enter null.

5 Column type: Enter the data type of the column. TEXT is entered by default. Alternatively, you can enter INTEGER, NUMBER, DATE, or DATETIME.

6 Column data: Configure a custom list transformation.

Information

Manual configuration of the list transformation is optional. In this case, the *Context* field is mandatory.

7 Context: Enter the data context of the transformation, e.g. with relation to the previous *steps* within your flow.

8 string: Enter the value you want to select.

Information

If you want to perform concatenation for the list transformation, you will have to make your entry in the JSONata editor. You can find more information under [Known issues > Concatenation in transformation fields](#).

- Example entry
 - *Title of table* - User list
 - *Column title* - User name
 - *Form field name* - PERSON.PERSONAL_DATA
 - *Column type* - TEXT
 - Intended transformation: Transform List<Person> into a list of names List<String>.
 - *Context* - stepBefore.persons
 - *string* - \$.name

Result definition via 'DynKwlColumnsResult'

Node

Finish

Properties



The result definition is empty.

The automation is executed synchronously. You can define the result object here.

DynKwlColumnsResult



Object properties

Title of table*

ab

j:



Columns The title of the table

Array

Input mode*

Basic entry



Columns*

Basic entry

Positions

List transformation

Result definition via *DynKwlColumnsResult* provides advanced configuration options. When configuring the columns, you can choose between the modes *Basic entry*, *Position*, and *List transformation*.

'Basic entry' input mode

Object properties

Title of table* ⓘ ab J: | v

Columns The title of the table Array

Input mode* v

Columns* ⓘ ab J: | v
Input required.

With the input mode *Basic entry*, you can customize the columns in the form of JSONata commands. Use the *Columns* field in the JSONata editor.

Example entry

```
//Example of a static variant for basic entry
[
  {
    "title": "Column1",
    "formFieldName": "Field1",
    "columnType": "TEXT",
    "data": [
      "Column1-Field1",
      "Column1-Field2"
    ]
  },
  {
    "title": "Column2",
    "formFieldName": "Field2",
    "columnType": "TEXT",
    "data": [
      "Column2-Field1",
      "Column2-Field2"
    ]
  }
]
```

'Positions' input mode

Columns The title of the table Array

Input mode* Positions ▼

+

Column title* ⓘ ab J: ▼

Form field name* ⓘ ab J: ▼

Column type* ⓘ TEXT ab J: ▼

Column data The column data Array

1 Input mode* Positions ▼

+

2 string 0 ab J: ▼

Information

The *Positions* mode corresponds to the setting options of the result definition via [SimpleDynKwColumnsResult](#).

Within the column data, you can also select the input mode (1) *Positions* and add string fields (2). Enter a fixed value in the string field. You can select preconfigured suggestions by selecting the arrow icon next to the field, for example.

'List transformation' input mode

The screenshot shows a configuration interface for a 'List transformation' step. It is divided into two main sections: 'Columns' and 'Column data'. The 'Columns' section is titled 'Columns The title of the table' and contains four fields: 'Input mode*' (set to 'List transformation'), 'Context*' (highlighted with a red box and labeled '1'), 'Column title*' (highlighted with a red box and labeled '2'), 'Form field name*' (highlighted with a red box and labeled '3'), and 'Column type*' (set to 'TEXT', highlighted with a red box and labeled '4'). The 'Column data' section is titled 'Column data The column data' and contains three fields: 'Input mode*' (set to 'List transformation', highlighted with a red box and labeled '5'), 'Context*' (highlighted with a red box and labeled '6'), and 'string' (highlighted with a red box and labeled '7'). Each field has a dropdown menu with 'ab', 'J:', and a downward arrow. A green 'Array' button is visible in the top right of each section.

The *List transformation* input mode focuses on the transformation of list data from previous steps of the flow.

1 Context: Define the context for the transformation, e.g. a previous step in the flow.

2 Column title: Enter the title of the column where the data from the transformed list should appear.

3 Form field name: Enter the name of the form field the data should be transferred to. If you don't want to transfer the data from this column to a form field, enter `null`.

4 Column type: Enter the data type of the column. `TEXT` is entered by default. Alternatively, you can enter `INTEGER`, `NUMBER`, `DATE`, or `DATETIME`.

5 Input mode: For further configuration, choose between the input modes *Positions*, *Basic entry*, and *List transformation*.

Information

Advanced configuration of the list transformation is optional. In this case, the *Context* field is mandatory.

6 Context: Enter the data context of the transformation, e.g. with relation to the previous *steps* within your flow.

7 string: Enter the value you want to select.

'Object created' trigger configuration



ELO objects and metadata

com.elo.flows/ELOBase/v1



Object created



Information

Form ⓘ

Field monitoring

Array

The *Object created* trigger starts when an ELO object (*SORD*) is created.

Advanced settings

1

Form ⓘ

2

Field monitoring Only run flow when all field monitoring is completed

Array

3

Check mode* ⓘ

4

Field name* ⓘ

5

Field value ⓘ

Optional: You can restrict the trigger to specific metadata forms and data fields that are in the context of the created object.

Information

You can use gen. 1 and gen. 2 ELO metadata. For more information on how to use metadata in ELO, refer to [General information > Using gen. 1 and gen. 2 metadata](#).

1 Form: Enter the metadata form the trigger listens to. The trigger is only started for this metadata form.

2 Field monitoring: Enable field monitoring if you only want to start the trigger within the context of specific metadata fields.

3 Check mode: Choose between the check modes *Field corresponds to a comparison value* and *Field was changed*.

- *Field corresponds to a comparison value* - triggers the flow if a specific value is set.
- *Field was changed* - triggers the flow only if a specific field has been changed.

4 Field name: Enter the date field you want to monitor.

5 Field value: If you select the check mode *Field corresponds to a comparison value*, enter the value to be checked.

Example entry

Form: Invoice (gen. 1) or BASIC.PERSON (gen. 2)

Check mode: *Field corresponds to a comparison value*

Field name: INVOICE_NO (index field) or INVOICE_POSITION.STATUS (aspect field)

Field value: 12345

Next steps

The trigger *Object created* is an asynchronous flow so you don't have to configure an end node.

'Object edited' trigger configuration



ELO objects and metadata
com.elo.flows/ELOBase/v1



Object edited



Information

Form ⓘ

Field monitoring

Array

The *Object edited* trigger starts when an ELO object (*SORD*) is edited.

Advanced settings

The screenshot shows a configuration interface for field monitoring. It consists of several numbered callouts pointing to specific elements:

- 1**: Points to the 'Form' input field.
- 2**: Points to the 'Field monitoring' section, which includes a toggle switch and a green 'Array' button.
- 3**: Points to the 'Check mode*' dropdown menu.
- 4**: Points to the 'Field name*' text input field.
- 5**: Points to the 'Field value' text input field.

Optional: You can restrict the trigger to specific metadata forms and data fields that are in the context of the edited object.

Information

You can use gen. 1 and gen. 2 ELO metadata. For more information on how to use metadata in ELO, refer to [General information > Using gen. 1 and gen. 2 metadata](#).

1 Form: Enter the metadata form the trigger listens to. The trigger is only started for this metadata form.

2 Field monitoring: Enable field monitoring if you only want to start the trigger within the context of specific metadata fields.

3 Check mode: Choose between the check modes *Field corresponds to a comparison value* and *Field was changed*.

- *Field corresponds to a comparison value* - triggers the flow if a specific value is set.
- *Field was changed* - triggers the flow only if a specific field has been changed.

4 Field name: Enter the data field you want to monitor.

5 Field value: If you select the check mode *Field corresponds to a comparison value*, enter the value to be checked.

Example entry

Form: Invoice (gen. 1) or BASIC.PERSON (gen. 2)

Check mode: *Field was changed*

Field name: INVOICE_NO (index field) or INVOICE_POSITION.STATUS (aspect field)

Field value: 12345

Next steps

The trigger *Object edited* is an asynchronous flow so you don't have to configure an end node.

Services

Services are actions that can be run within a flow. The *ELO objects and metadata* component offers the following services:

'Permissions' group

Name	Function
Remove all permissions	Removes all permission entries from an object
Remove permissions	Removes permissions from an object
Add permissions	Adds permissions to an object

'Documents' group

Name	Function
Download working version	Downloads a version of an object and provides it in flows
Upload document version	Loads a FlowsFile in an object as a document version

'Links' group

Name	Function
Create ECD link	Creates a new ECD link based on an object ID of a document
Create external link	Creates a new external link based on an object ID

'Metadata' group

Name	Function
Load metadata	Reads the metadata of an object and provides it in the flow
Change color	Sets the color of an object
Change metadata form	Sets the metadata form of an object
Set metadata	Sets defined metadata on an object

'New' group

Name	Function
New documents	Creates several new documents
New document	Files a new document to the repository
New folder	Creates a new folder in the repository

'Organize' group

Name	Function
Delete all references	Deletes all references to an object
Copy	Copies an ELO object and files the copy
Delete	Deletes an ELO object
Move	Moves an entry in the repository
Reference	Creates a reference to an ELO object in the repository

'Additional metadata' group

Name	Function
Load map fields as list	Loads map fields of an object as a list
Load map fields	Loads map fields of an object and provides them in flows
Write map fields	Writes values to map fields for an object.

Without group assignment

Name	Function
Load configuration file	Loads a configuration file from the ELO repository

ELO search



Goal and use

ELO search offers services that can be used to search for SORDS in an ELO repository.

Credentials and connection data

Configure connection data for the *ELO search* component. A *default connection* is available. This is entered in the config files. The *user context* assumes the session of the user logged into the ELO client. You can also create additional connections to run services with defined rights.

Triggers

Triggers are the starting points of a flow. The *ELO search* component offers the following trigger:

Name	Function
Perform form search	With this multi-instance trigger, you can search for SORDs that are assigned to a specific metadata form. It is also possible to filter for the contents of aspect fields.

Services

Services are actions that can be run within a flow. The *ELO search* component offers the following services:

Name	Function
Child search (metadata generation 1)	Search for SORDs located at the repository level under the start SORD. This service only finds SORDS with generation 1 metadata.
Child search (gen. 2 metadata)	Search for SORDS located at the repository level under the start SORD.
Metadata form search (gen. 2 metadata)	Search for SORDs assigned to a specific metadata form. It is also possible to filter for the contents of aspect fields.

ELO feed



Goal and use

The *ELO feed* component offers basic functions for creating and editing feed entries.

Credentials and connection data

Configure connection data for the *ELO feed* component. A *default connection* is available. This is entered in the config files. The *user context* assumes the session of the user logged into the ELO client. You can also create additional connections to run services with defined rights.

Triggers

Triggers are the starting points of a flow. The *ELO feed* component does not offer any triggers.

Services

Services are actions that can be run within a flow. The *ELO feed* component offers the following services:

Name	Function
------	----------

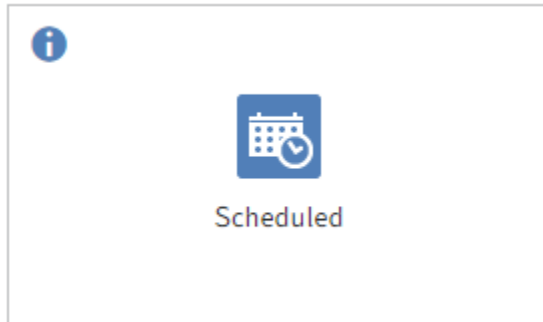
Create post	Writes a feed entry to an object
-------------	----------------------------------

Use

The *Create post* service accepts a string as a parameter. Use the following format to use a hashtag or mention:

- #[HASHTAG]
- @[User]

Time



Goal and use

The *Time* component allows you to schedule start times for flows.

Credentials and connection data

This component does not require credentials.

Triggers

Triggers are the starting points of a flow. The *Time* component offers the following triggers:

Name	Event
Scheduled trigger	Started via cron expression.

Use:

Define when to start the flow via a cron expression. ELO Flows uses the *Spring* cron format with a seconds field. This expression has six fields.

Structure

	Second	Minute	Hour	Day of month	Month	Day of week
Mandatory	YES	YES	YES	YES	YES	YES
Permissible values	0-59	0-59	0-23	1-31	1-12 or JAN-DEC	1-7 or MON-SUN
Permissible characters	, - * /	, - * /	, - * /	, - * ? / L	, - * /	, - * / L

Please note

Fields cannot be allocated twice. Double allocation such as 0 * * * * * does not work here.

You can increase the complexity of the cron expression with the following characters:

Character	Meaning	Example
*	All values	* in the minutes field = Every minute
?	No specific value	0 0 * ? * * = Every hour, no matter the day
-	Range between two values	0 0 10-12 ? * * = Every hour between 10:00 and 12:59
,	Additional values	0 10,44 14 ? * * = At 14:10 and 14:44
/	Incremental values	0 0/5 * * * ? = Every 5 minutes
L	Last value	* * * L * ? = Every second on the last day of the month

Examples

Cron expression	Meaning
0 0 * ? * *	Every hour
0 * * ? * *	Every minute
0 * 14 * * ?	Every minute, between 14:00 and 14:59
0 0/5 14 * * ?	Every 5 minutes, between 14:00 and 14:59
0 */5 20-23,0-7 ? * 1-6	Every 5 minutes from 20:00 until 23:59 and 00:00 until 07:59, Monday to Saturday
0 10,44 14 ? 3 WED	At 14:10 and 14:44, only every Wednesday, only in March
0 15 10 ? * 6L	At 10:15, on the last Saturday of the month
0 11 11 11 11 ?	At 11:11, on day 11 of the month, only in November
0 0 12 1/5 * ?	At 12:00, every 5 days
0 15 10 L-2 * ?	At 10:15, 2 days before the last day of the month
0 15 10 L * ?	At 10:15, on the last day of the month
0 15 10 15 * ?	At 10:15, on day 15 of the month

Services

Services are actions that can be run within a flow. The *Time* component does not offer any services.

ELO workflow



Goal and use

The *ELO workflow* component offers functions for starting an ELO workflow from within a flow. Flows can also be triggered by workflows.

You will find information on ELO workflows and using the workflow designer under [Administrators > Processes and automation > ELO workflow](#).

Credentials and connection data

Configure connection data for the *ELO workflow* component. A *default connection* is available. This is entered in the config files. The *user context* assumes the session of the user logged into the ELO client. You can also create additional connections to run services with defined rights.

Triggers

Triggers are the starting points of a flow. The *ELO workflow* component offers the following triggers:

Name	Event	Additional information
Workflow node event	This trigger can be used to start a flow from a workflow node.	Synchronous flow
Workflow node direct call	This ensures that only the defined flow is started.	Synchronous flow

'Workflow node event' trigger usage



ELO workflow

com.elo.flows/ELOWorkflow/v1





Workflow node event

ELO workflows allow you to control and map business processes.



Information

Event* 

exampleEvent 

In the flow designer, select the trigger *Workflow node event*. Under *Event*, enter a name. For example, *exampleEvent*.

In the ELO workflow designer, select a node you want to trigger a flow with.

The ELO workflow component enables integration of ELO Flows in ELO workflows.

Additional information regarding the use of flow nodes in the workflow designer can be found under [ELO workflow > Default workflow > Flow node](#).

'Workflow node direct call' trigger usage



ELO workflow

com.elo.flows/ELOWorkflow/v1



Workflow node direct call

ELO workflows allow you to control and map business processes.



Information

In the ELO workflow designer, select a node you want to trigger a flow with.

Services

Services are actions that can be run within a flow. The *ELO workflow* component offers the following services:

Name	Function
Start workflow	Starts the defined workflow on an ELO object
Start workflows	Starts each defined workflow on multiple transferred ELO objects
Read workflow map fields	Reads map fields of a workflow
Write workflow map fields	Writes values to map fields of a workflow

SMTP Mail



SMTP

Goal and use

The *SMTP Mail* component allows you to send e-mails (as HTML or text) with the help of a mail server.

Credentials and connection data

This component requires access to an SMTP server.

Triggers

Triggers are the starting points of a flow. The *SMTP Mail* component does not offer any triggers yet.

Services

Services are actions that can be run within a flow. The *SMTP Mail* component offers the following services:

Name	Function
Send e-mail	Sends a text e-mail
Send e-mail (HTML)	Sends an HTML e-mail (this service allows you to apply different formats using HTML tags in the message text)

FTP



FTP

Goal and use

The *FTP* component provides functions for communicating with an FTP server. It enables you to perform folder and file operations, such as uploading and downloading documents to and from an FTP server or loading data and metadata.

Credentials and connection data

Configure connection data for the *FTP* component. Each connection entry stands for a connection to the FTP server. Different FTP protocols can be used: insecure FTP, FTPS, and SFTP.

Triggers

Triggers are the starting points of a flow. The *FTP* component does not offer any triggers.

Services

Services are actions that can be run within a flow. The *FTP* component offers the following services:

'Document' group

Name	Function
Download document	Downloads a document from an FTP server. This allows the document to be used for other services.
Load metadata for document	Reads the metadata of a document on an FTP server. This allows the available metadata, such as the path, size, and timestamp to be used for other services.
Upload document	Stores a document on an FTP server.

'Folder' group

Name	Function
Create directory	Creates a new folder on an FTP server. It is also possible to create multiple child directories.
Load metadata for folder	Reads the metadata of a folder on an FTP server. This allows the available metadata, including path, size, and timestamp to be used for other services.

Name	Function
Load directory contents	Reads the content of a folder. You can also restrict or expand the selection with different filters and options.

'Organize' group

Name	Function
Copy	Copies a document or folder to an FTP server. This creates a duplicate, which may be stored at a different location on the FTP server.
Delete	Removes a document or folder on an FTP server. All contents in the folder or document will also be deleted permanently. You cannot delete the root directory of the FTP server.
Move	This service moves a document or a folder on an FTP server to another location on the FTP server. You cannot move the root directory of the FTP server. You also have the option to simultaneously rename the document/folder.
Rename	Renames a document or folder on an FTP server. The typical name restrictions apply.