UNICORN: A USER-ORIENTED DOCUMENTATION

# Covered Topics

[Covered Topics 1](#_Toc6054834)

[Overview of the navigation options 1](#_Toc6054835)

[Important file types and their meanings 2](#_Toc6054836)

[Introduction in some frequently used use cases 2](#_Toc6054837)

[HOW-TO: Register an event type 2](#_Toc6054838)

[HOW-TO: Export an event type to file 2](#_Toc6054839)

[HOW-TO: Register one or several event(s) 2](#_Toc6054840)

[HOW-TO: Set and manage rules for attribute dependencies 3](#_Toc6054841)

[HOW-TO: Manage external rules for attribute dependencies 3](#_Toc6054842)

[HOW-TO: Manage the event repository 3](#_Toc6054843)

[HOW-TO: Work with Live Queries 3](#_Toc6054844)

[HOW-TO: Register a user account 4](#_Toc6054845)

[HOW-TO: Log-in to your user account 4](#_Toc6054846)

[HOW-TO: Register a notification rule 4](#_Toc6054847)

[HOW-TO: Delete a notification rule 4](#_Toc6054848)

[HOW-TO: Manage arrived notifications 4](#_Toc6054849)

[Known Issues 5](#_Toc6054850)

[Distinctive features of the UI – as at April 13](#_Toc6054851)[th](#_Toc6054851) [5](#_Toc6054851)

[Suggestions to improve the UI– as at April 13](#_Toc6054852)[th](#_Toc6054852) [5](#_Toc6054852)

[Known functionality issues – as at April 13](#_Toc6054853)[th](#_Toc6054853) [5](#_Toc6054853)

# Overview of the navigation options

|  |  |  |
| --- | --- | --- |
| tag | sub tag | functionalities |
| Home |  | Return to Unicorn’s start view |
| Import | XML/XLS/XSD | Import files where events or event types saved structured |
| BPMN |  |
| Replayer | Simulate a specific sequence of events |
| Generator | Generate several events  Set, manage, import and export attribute dependencies |
| Adapter | Get events from a supported source |
| Processing | Event Transformation |  |
| Event Correlation |  |
| Event Repository |  | Manage/ view registered events and event types  Create and export event types  Manage/ view registered process instances |
| Queries | Live | Create, manage and monitor live queries |
| Monitoring | BPM |  |
| Attribute Charts |  |
| Event Views |  |
| Notifications |  | Set and manage notification rules |
| Export |  | Export several events of an event type to supported files |
| Sign In |  | Create user accounts and log-in to an existing account |

# Important file types and their meanings

|  |  |
| --- | --- |
| FILE TYPE | IMPORTANCE/ MEANING |
| .json | * represent recorded attribute dependencies * represent recorded event types |
| .xsd | * represent recorded event types |
| .csv | * represent recorded sequence of events of one event type |
| .xml | * represent recorded event of one event type |
| .zip | * represent an archive of several events, each one exists as xml-file in the zip-archive |

# Introduction in some frequently used use cases

## HOW-TO: Register an event type

1. The event type is already descripted in a file

* You can upload the .xsd file to Unicorn by using the import functionality which you can find by “Import” 🡪 “XML/ XLS/ XSD”.

1. You have to model the new event type

* By following “Event Repository” 🡪 “Event Type” 🡪 “Create event type” you open a dialog where you can characterize the new event type.

## HOW-TO: Export an event type to file

There is only an export functionality for event types in “Event Repository” 🡪 “Event Type”. Search, maybe with the help of the filter methods, the event type which you want to export and press the button who passed to your desired file type and the right event type.

## HOW-TO: Register one or several event(s)

Before it is possible to register an event, you have to proof that the related event type is already registered. If not, follow the description for registering an event type.

There are some ways to register events.

1. Read one or more events from a supported file

Go to “Import” 🡪 “XML/ XLS/ XSD”. Drag and drop or select a supported file and press finally “Upload”.

1. Generate several events based on given bounds

Go to “Import” 🡪 “Generator”. Select which event type you want to generate and the amount, too. Below of these settings, Unicorn has loaded all attributes of the current selected event type. Give for each attribute possible or wanted values which are separated by “;” from each other.

1. Replay a specific sequence of saved events

If you have a sequence of events which you want to replay in their temporally order, you can use the functionalities of the “Replayer”.

## HOW-TO: Set and manage rules for attribute dependencies

Rules for attribute dependencies helps you to define how value of an attribute depends to the value of another attribute. If you use the functionality to generate several events, the attributes values will be districted by the defined dependencies.

Go to “Import” 🡪 “Generator”. Select “Attribute dependencies”. Select the corresponding event type and the attributes who depend on each other.

If you press “Add dependency”, Unicorn create the dependency in its database. Below, you can add or delete the concrete entries. If Unicorns already knows about the dependency because you added some entries or only the dependency in the past, Unicorn will show you all already existing entries.

It is possible to delete a dependency, too. Therefor press “Delete selected” instead of “Add dependency”. Note, that it is not possible to remove an attribute dependency, if there any corresponding entry which describes the concrete values. Please, remove each remaining entry before you delete the abstract dependency.

## HOW-TO: Manage external rules for attribute dependencies

Go to “Import” 🡪 “Generator” 🡪 “Export/import dependencies”. Choose the event type whose attribute dependencies you want to export and press “Export”.

The same view provides the functionality to import such previously created file.

## HOW-TO: Manage the event repository

If you have to manage the registered events or the defined event types, you will find both functionalities close to each other. There’s no matter of fact, if you manage an event or an event type. Simply change your view between “Event” and “Event Type” when you go to “Event repository”.

Such a view provides a list of all objects of the managed type. You can easily filter by using the dropdown menu and specify necessary filter parameters or reset all defined filter option by pressing “Reset”. There is the task to remove one, some or all items of the object type? Select one, some or press “Select All” and press “Delete” to full fill this task.

## HOW-TO: Work with Live Queries

First, you have to take care about the difference between live queries and queries which you can ask in the event repository. Live queries are supported by the Esper engine and they are a fundamental aspect for notification rules. It is possible to name the queries to the event repository ‘static queries’ because they only search in the amount of already registered events. On the other hand, live queries proof every event that arrive if it satisfies the query conditions.

To work with live queries, you have to move to UI area “Queries” 🡪 “Live”. Each query has a name and a statement which you can manage with the help of the forms “Query name” and “Query statement”. The form “Saved queries” provides a list with the queries which are already registered. Select an entry to manage the corresponding query. The “show”-button reopen the possibility to change and check the statement. If you want to see all arrived events that were filtered by the corresponding query, use the “Show Log”-button.

The syntax of the query statements orientates itself by the syntax of SQL queries. To get some tips, use the “Help”-button or check out the documentation of the Esper engine.

Let’s look to one example! We define an event type named “*Transaction*”. It has the following attribute signature:

*{transactionId: String; customerName: String; cardStatus: String; creditAmount: Float; purpose: String; location: String}*

Our task is to filter out continuously all arriving transaction events who are produced at location “xtown”. We are interested to the name of the customer who did the transaction, but the amount of the credit, too. To handle this task, let’s create a new live query with a suitable name. Enter following query statement:

*SELECT creditAmount, customerName FROM Transaction WHERE location=’xtown’*

Now it’s easy to associate the given solution with the given problem. Note that you can create much more complex queries and patterns!

## HOW-TO: Register a user account

User accounts are the second fundamental aspect of notification rules besides live queries. To add a new user account to the Unicorn service, move to “Sign in” and select “Register here”. Choose a password, an user name and enter your email address. Finish the process by using the “Register”-button.

## HOW-TO: Log-in to your user account

Move to “Sign in”, enter your email and your passport. Confirm your account credentials with the “Login”-Button. Do not forget to log-out after your work at Unicorn has finished!

## HOW-TO: Register a notification rule

Before you deal with notification rules, please get insides of user accounts and live queries. Therefore, you may read the sections “[HOW-TO: Register a user account](#_HOW-TO:_Register_a)” and “[HOW-TO: Work with live queries](#_HOW-TO:_Work_with)”.

To register a notification rule, move to “Notifications” and press the Button with the caption “Add notification rule”. Choose a priority of the notification rule. After that, select the user who will receive the notification. Finish the process by decide which live query implements your desired rule. Confirm finally by pressing the “Create”-button.

Event time an event arrived Unicorn, respectively to the Esper engine, it is checked if it matches to the query condition. If the check equals to true, the user will receive a corresponding notification.

If you registered several rules, the priority decides the order in which they will be handled.

## HOW-TO: Delete a notification rule

The action of deleting a notification rule is as simple as to creating one. In the same environment where you created a notification rule, you can find a list of all registered notification rules. Press “Delete all notification rules” to delete all or “Delete” behind one rule to delete only this one.

## HOW-TO: Manage arrived notifications

To see and manage the arrived notifications, please log-in to your user account. If you need any help, read the section “[HOW-TO: Log-in to your user account](#_HOW-TO:_Log-in_to)”!

You can manage your notifications if you move to “Notifications”. It’s the same interface to manage your arrived notification as you already known from the event repository. To unpin a notification, select “Seen” instead of “Delete” in the event repository.

# Known Issues

## Distinctive features of the UI – as at April 13th

* You have to choose „Add dependency” if you want to register an attribute dependency, but if you want to edit already registered dependencies, too.
* A user doesn’t get any descriptions or tips how he or she could work with the graphical user interface. There is only a button that helps you, if you want to work with the queries.
* If you want to import an event or an event type, both functionalities are close to each other. The “Import”-Menu its supports. But for the other direction, to export an event or an event type, the functionalities aren’t close to each other. It’s the same issue for creating manually an event or an event type. Maybe it’s more user-optimised if the “Import”-functionalities and the “Export”-functionalities manage the same features but in different directions.
* There isn’t a feature to reset the instance of unicorn.
* The “Home”-view doesn’t provide any features or information.
* It’s difficult to edit or view attribute dependencies because you have to know which attributes are already connected.
* The output form of live queries is extremely small. Therefore, it’s difficult to read the log, but to resize the form, too.
* A function to remove user accounts is missing.

## Suggestions to improve the UI– as at April 13th

* A documentation or something like a wiki provided by the Unicorn-UI could be fine. Provide support buttons, like this for live queries, for other features to support the user interaction.
* Maybe it could be used for instructions or as a list of frequently used or important features.
* Standardise the functionalities of import and export. Extract and combine features of the same concepts.
* Add some features to reset the instance of Unicorn
* A feature to show all attribute dependencies of an event type would be fine.
* There is enough space to provide a form with a bigger standard size to output a query log.
* Add the function to remove user accounts.
* A new interface besides the existing interfaces “Home”, “Event repository” and so on may be good to reset the current instance of Unicorn, remove user accounts or to edit properties like these for email based notifications.

## Known functionality issues – as at April 13th

* The event Replayer does not import the attribute values of the events from csv-files as well as from zip-archives.