RAYVENTORY®

Technology Asset Inventory

RayVentory Data Hub Administration and User Guide 12.5

raynet

RayVentory Data Hub Administration and User Guide RayVentory Data Hub 12.5

Raynet GmbH Germany
See our website for locations.

www.raynet.de

Contents

Introduction

RayVentory Data Hub is a very versatile data collection and reporting tool that supports the creation of reports and dashboards without requiring in-depth knowledge of the origin database structure. RayVentory Data Hub empowers companies or management to derive high quality information that is a prerequisite for making informed decisions.

This guide shows how to configure and manage RayVentory Data Hub, so that IT departments can stay ahead, save time, increase productivity, and decrease IT costs. RayVentory Data Hub is based on the client-server architecture in which all the information, data, and configuration is stored on the RayVentory Data Hub server. Users work on this server remotely through the RayVentory Data Hub web-based client. Data collected through the Data Hub Agent is also stored on the RayVentory Data Hub server but into a second database, hereinafter referred to as "Result"-database. This database stores the raw data collection results, which can and will be used to generate reports and dashboards that are bound to this result data.

About This Guide

This document has been written to guide both potential users and administrators of RayVentory Data Hub by describing all of the important features and functions of the RayVentory Data Hub web-client. Some of the topics in this document refer to other documents which are delivered with this product. In case a user needs more information or wants to contribute towards the improvement of this document, he can reach us by using our customer support channel.

Manual Conventions

The following typesetting conventions are used in this manual:

- Cross references to other manuals are shown in italics:
  "This can be found in the *RayVentory Data Hub Release Notes*."
- Cross references and external links are shown in blue and are underlined:
  "See RayVentory Data Hub for..."
- Quotations from the computer screen (titles, prompts, and so on) are shown in bold:
  "Go to **Devices** screen."
- Code syntax, file samples, directory paths, entries that you may type on screen, and the like are shown in a monospaced font:
  "Use `docker compose -up` to set your instance up"
- Large blocks of code are shown in a monospaced font with a grey background:

```
version: "3.7"
services:
```

- Italics may also be used for emphasis: "This manual is *not* intended..."
- Bold may also be used for inline headings: "**Target**: Indicates a target frame..."

Two note formats are used in RayVentory Data Hub documentation

This is the basic format for giving additional information to the current topic. It can come with four different headings:	
	**Be aware:** This note format contains important information related to your current activity. You should not skip over this text.
	Note: This format is used for items of interest that relate to the current discussion.
	**Best practice:** If there is a best practice approach to the current topic you can decide if you want to follow it, or stick to your own plan.
	**Tip:** Tips are designed to help you find the easiest and quickest way to work with RayVentory Data Hub.

The second format is for very serious alerts.

	**WARNING** The information here may save you from data loss. Pay particular attention.

Documentation Requiests

We welcome suggestions and input on the various documentation resources available for RayVentory Data Hub and its components. Feedback and other concerns can be forwarded through local Raynet support representative.

Features

RayVentory Data Hub is a data collection and reporting-tool for the creation of standardized reports and dashboards for the own application lifecycle management platform. Some of the salient features of RayVentory Data Hub include:

- Collecting data from SQL-Servers, OBDC and Excel, SaaS platforms, Active Directory, and much more
- Customizable and unlimited task scheduling, per registered Data Hub Agent, for unattended background execution,
- Creating, editing and publishing reports and dashboards,
- Configurable user roles (application wide) and permissions per report and dashboard,
- Multi-tenant support, with secure data encryption

RayVentory Data Hub is based on a client-server architecture in which all report and dashboard

information, as well as data and configuration is stored on the RayVentory Data Hub server. Users work on this server remotely through its web interface, for example by using the web browser of their choice.

Core components include:

- **Data Hub** (server)
  - The backend - an .NET web application, exposing HTTP port (default `8090`)
  - The frontend - a web-based app, used to control all aspects and settings, as wells as view and edit reports
- **Data Hub Agent** (client) - an installable client part, serving as an agent, providing scanning capabilities and scheduled execution. The agent can address any available source, both local and remote, including SQL databases, Active Directory etc. The agent may, but does not have to be installed on the same machine as the backend - in fact, it is perfectly possible and sometimes desirable to have more agents installed on various machines, in order to:
  - Ensure that all data sources are accessible
  - Provide an extra level of parallelism

For a complete configuration, Data Hub server and at least one Data Hub Agent are required. Both components must be installed separately by their respective Windows Installer setups.

Data Hub

This section of the document describes the RayVentory Data Hub web-client in detail. After going through this chapter, a user is supposed to be fully aware of the features offered by this product and have a technical know-how on how to work with those features.

Prerequisites

Hardware Requirements

Requirements when SQL Server and RayVentory Data Hub are installed on the same machine:

- Min. 4 CPU cores
- Min. 8 GB of RAM
- Min. 20 GB of disk space

Requirements when only RayVentory Data Hub is installed on the machine:

- Min. 4 CPU cores
- Min. 4 GB of RAM
- Min. 10 GB of disk space

Software Requirements

The following are the minimum software requirements for the installation and running of RayVentory Data Hub:

- Microsoft Windows Server 2012 R2 or higher
- IIS 8 or higher
- Microsoft .NET 6.0 – Windows Server Hosting Bundle (https://dotnet.microsoft.com/en-us/download/dotnet/6.0)
- Microsoft SQL Server 2016 or SQL Server Express 2016
- If RayVentory Data Hub Agent is installed on the same machine as the server, then all requirements of Data Hub Agent also apply

Note:
In order to run hosting bundles, the "Universal C Runtime" is required. Modern Windows Servers should already have it, but it may be required to download for older ones. The oldest supported OS is currently Windows Server 2012 R2. More information can be found here: https://support.microsoft.com/en-us/help/2999226/update-for-universal-cruntime-in-windows

Supported Web Browsers

- Microsoft Edge version 80 and newer
- Mozilla Firefox version 74 and newer
- Google Chrome version 80 and newer

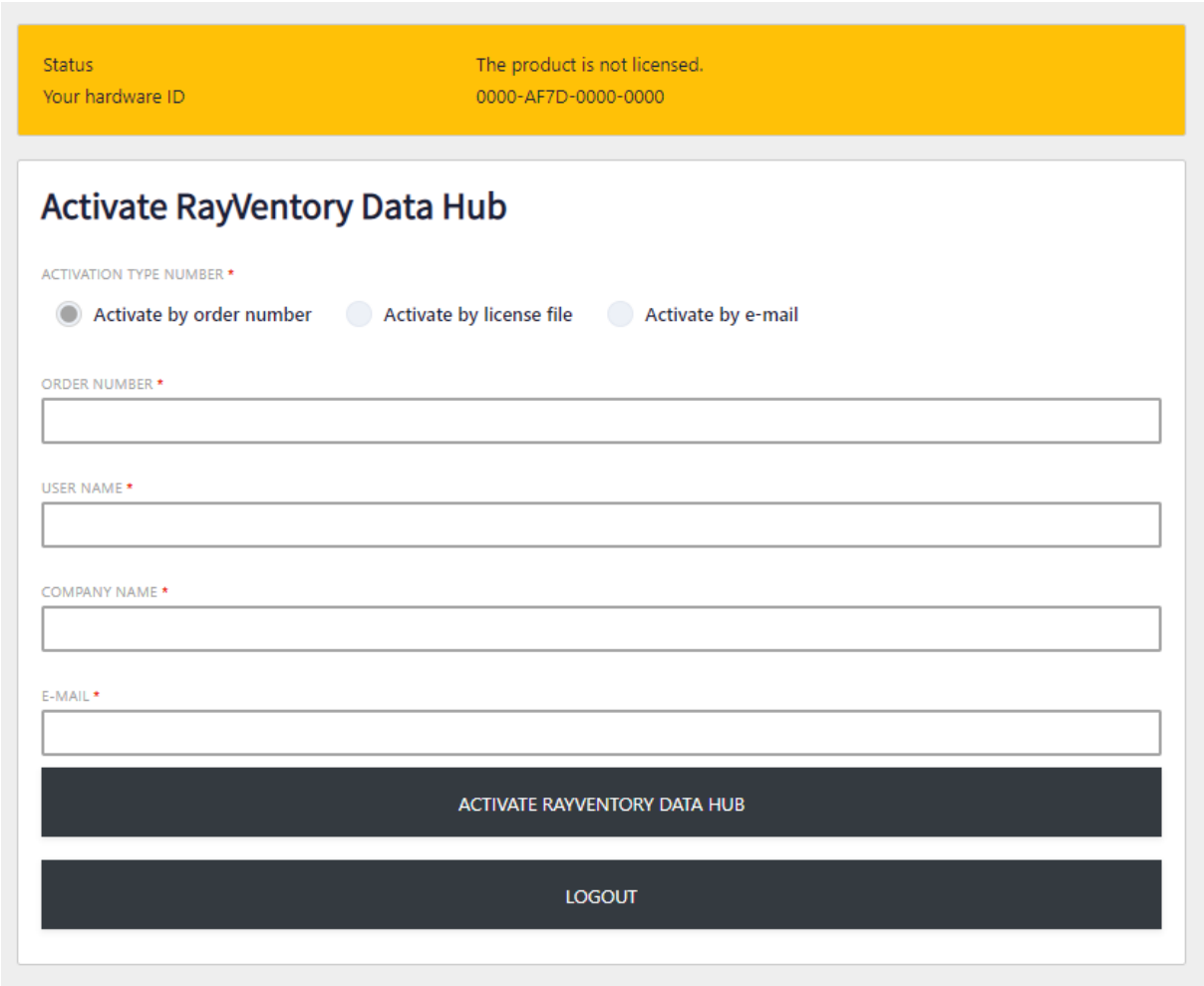
Getting Started

RayVentory Data Hub provides the ability to create and customize reports and dashboards the way you want, filled with customized data that you need. Schedule your own tasks to collect data from multiple different sources, including SQL-Server, OBDC, Excel, Active Directory, Microsoft Graph API and other sources. Each task execution creates a new table or updates existing ones within RayVentory Data Hub result database. The tables are filled with the data collected from your specified query. This data can then be used by binding it to the web report- and dashboard-tool provided by RayVentory Data Hub . Further details about scheduling and defining queries are given in the sections Tasks.

Dashboards are a simple way to organize and manage multiple charts that are bound to the underlying data stored in RayVentory Data Hub 's result database. Dashboards contain one or more dashboard items that give you an overview of the dimensions and metrics you care about most. Reports cover a wide range of topics, but usually focus on transmitting information with a clear purpose. The transmission of information is supported by the reporting-tool through a rich set of layouting tools that meet business needs.

License

RayVentory Data Hub requires a license. Depending on the installation type, the product may have already been set-up with a correct license, or otherwise an initial screen will be shown when accessing an instance without a license.



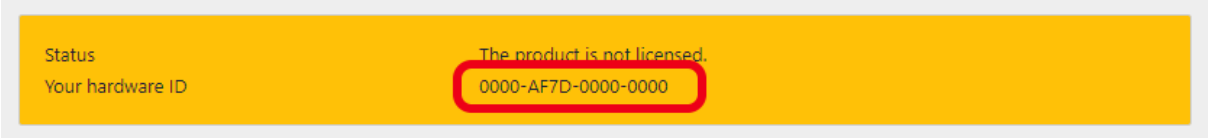
Note:
The product can be only activated by the site administrator / root account. The other users, including project administrators, will only see a prompt about a missing license, but they will be unable to activate it. If you see the message about a missing license, but you are not the site administrator or you are not authorized to sign-in as a site administrator, contact your administration to perform the activation.

If you see the message about a missing license and you have another account which belongs to the group of site administrators, press **LOGOUT** and sign-in again using site administrator credentials, in order to be able to activate the product.

The product can be activated using one of the following methods. They all at the end make sure that the product is in the activated state, and the selection of the method will be based on the type of the license received from Raynet. When selecting the right activation method, bear in mind that the **activation by order number** requires a one-time internet connection to Raynet activation server.

If there is no license available yet, contact Raynet support for assistance. Provide the **HARDWARE ID** of the device on which the product is to be activated. The information about the

current hardware ID can be found on the top of the screen (if signed-in as a site administrator), for example:



Hardware ID is a special number which is unique for each machine, but it does not identify it, and cannot be used to determine any physical or software evidences. It is used by Raynet to ensure, that once a license is activated it cannot be transferred to another machine.

Activation by Order Number

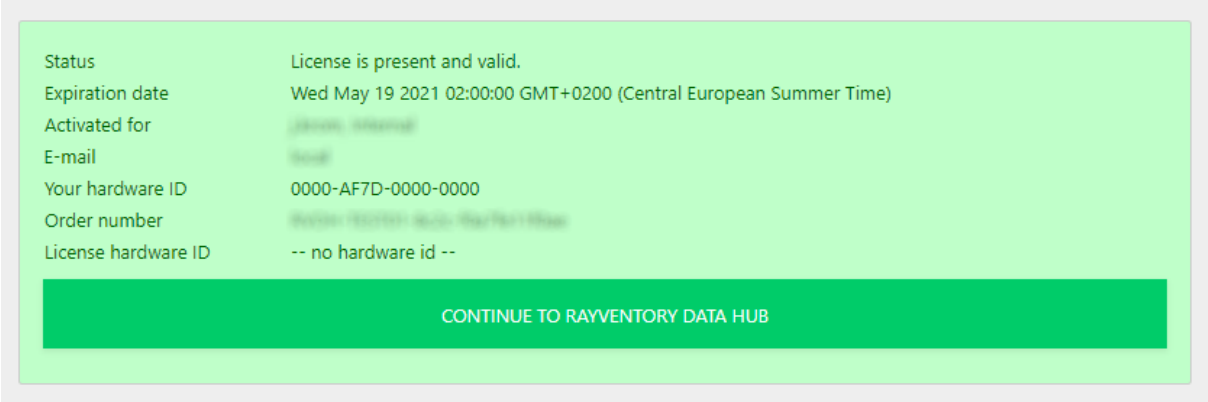
Use this method, if an order number is available. The order number is a string, consisting of letters, digits and hyphens. When using this method, the following details are required:

- **User name**
  The actual user, name, or any value that will help to identify who activated the product.
- **Company name**
  The name of the company or division where the product is being activated.
- **E-mail**
  The e-mail address of a contact person.

After providing the required data, press **ACTIVATE RAYVENTORY DATA HUB** to start online activation process.

Note:
This activation method will do a one-time connection to Raynet activation server to validate the license. Ensure, that the host machine where RayVentory Data Hub is hosted is able to connect to the internet. The connection to the server is using a TLS/SSL connection on a default SSL port.

Once the activation server responds with a data, the license will be saved on RayVentory Data Hub server, and a confirmation will be displayed. The product can be used from now on.



Pressing **CONTINUE TO RAYVENTORY DATA HUB** to continue to the Home view.

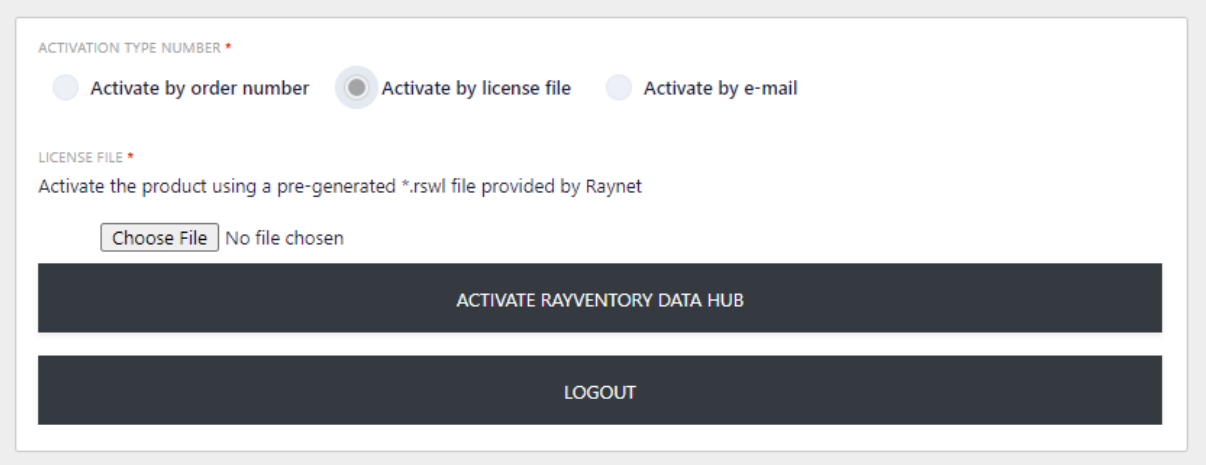
Activation by License File

Use this method, if a license file in any of the following formats is available:

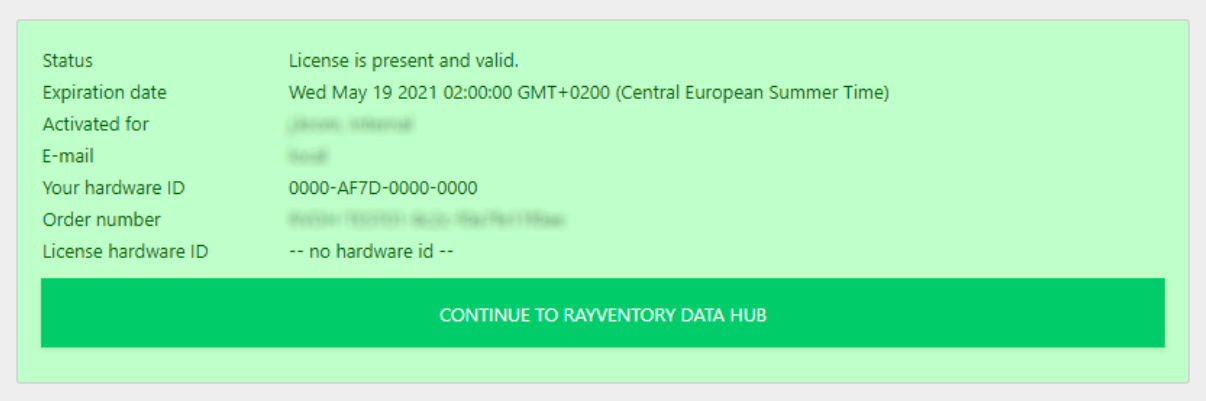
- Raynet license file (`*.rswl`)
- Raynet bundled license file (`*.rslx`)

> **Note:**
> Classic Raynet desktop licenses (with extension `*.rsl` or `*.rsfl`) are not supported.

In order to activate, select **Activate by license file** radio button, and then press **Choose file** to open the license picker dialog.



After pressing **ACTIVATE RAYVENTORY DATA HUB**, the license file will be transferred to the RayVentory Data Hub server and validated. No internet connection with external servers or internet is required, the validation is performed in an offline mode. Once the license is validated, the process will be automatically finished and a confirmation will be shown. The product can be used from now on.



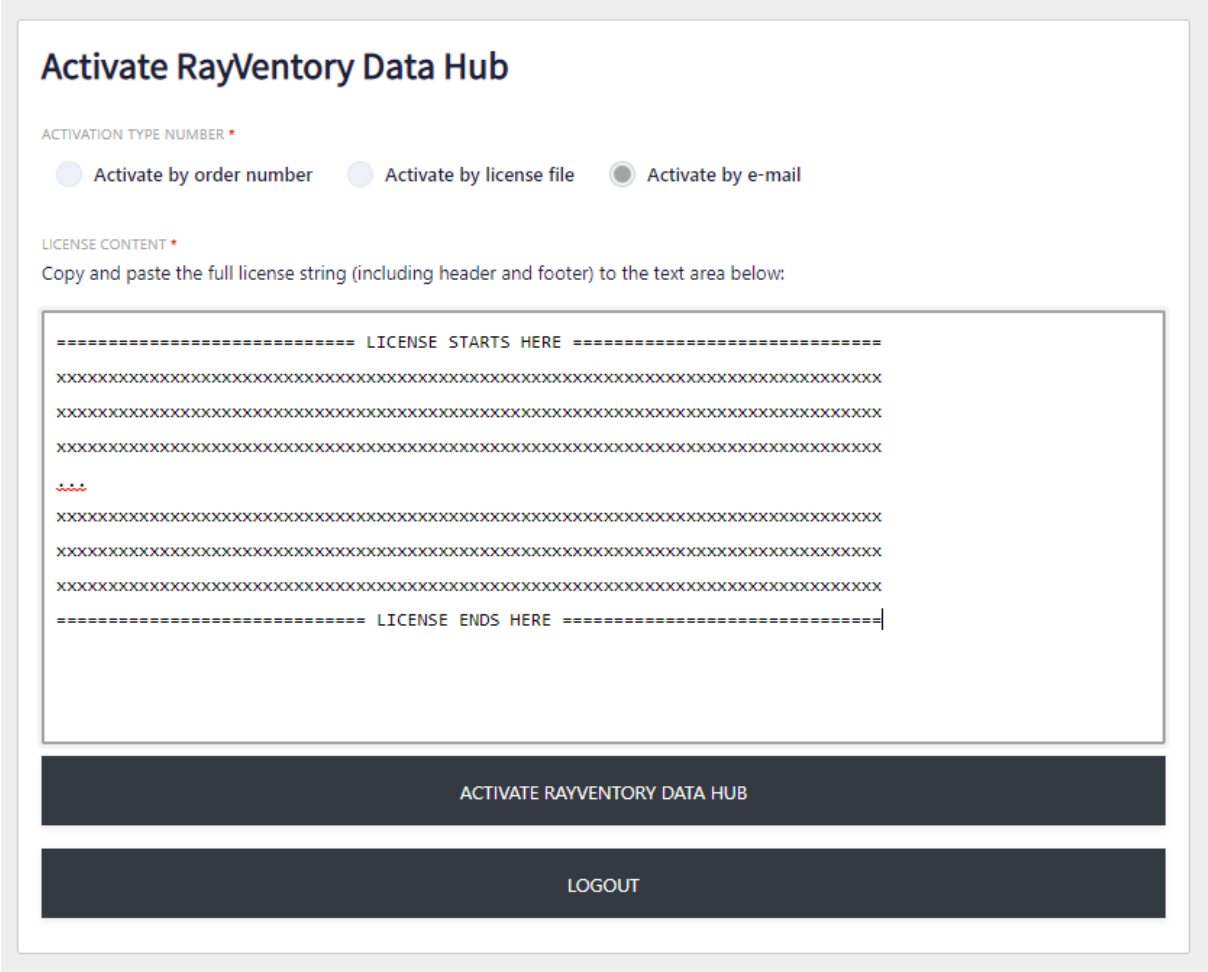
Pressing **CONTINUE TO RAYVENTORY DATA HUB** will bring you to the Home view.

Activation by E-mail

Use this method, if you used an e-mail based licensing. The offline activation performed by Raynet support will result in a license content sent in a plain e-mail. The license string contains digits, letters and some special characters, and starts and ends with a special header, for example:

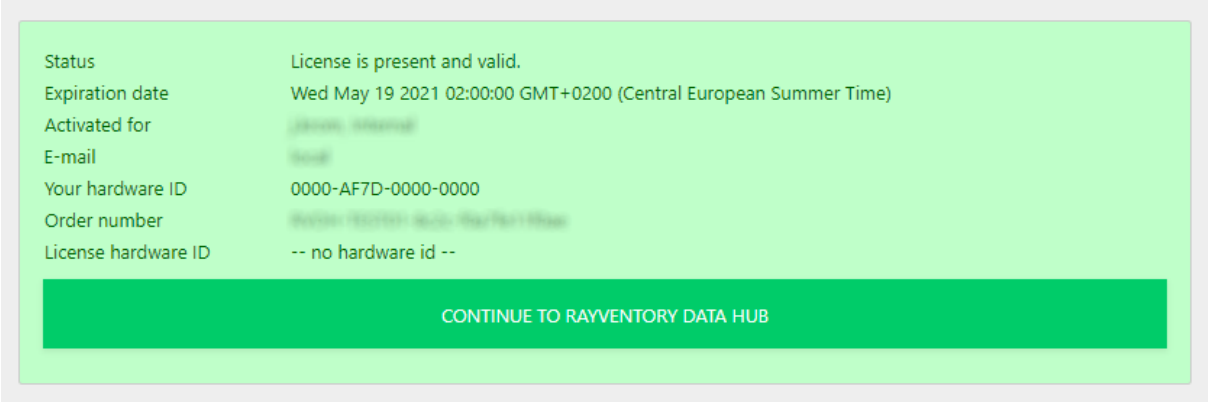
```
============================== LICENSE STARTS HERE ==============================
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
...
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
============================== LICENSE ENDS HERE ==============================
```

Just copy the whole string, including the opening and closing lines and put the whole content into the respective field.



After pressing **ACTIVATE RAYVENTORY DATA HUB**, the license data will be transferred to the RayVentory Data Hub server and validated. No internet connection with external servers or

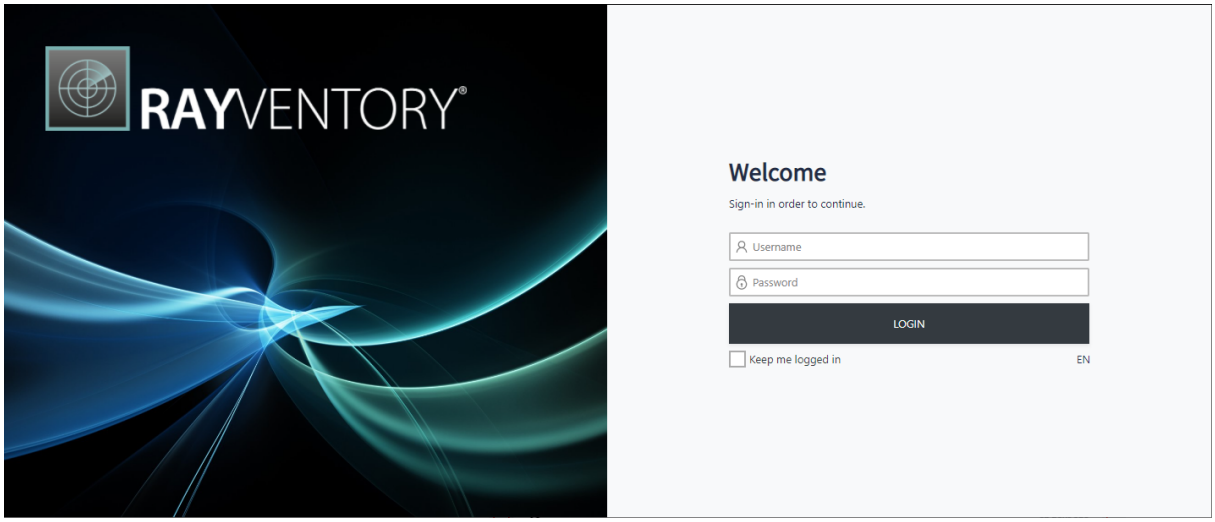
internet is required, the validation is performed in an offline mode. Once the license is validated, the process will be automatically finished and a confirmation will be shown. The product can be used from now on.



Pressing **CONTINUE TO RAYVENTORY DATA HUB** will bring you to the Home view.

Login

Once the RayVentory Data Hub Server is installed on a machine, launching the Web browser and navigating to the Server's URL will navigate to the login screen of RayVentory Data Hub.



Enter your username and password and confirm them by pressing the **LOGIN** button. After a successful log-in the page redirects to the Home screen. You can also configure RayVentory Data Hub to remember your credentials, by selecting **Keep me logged in** before signing in. This will remember the session for next 7 days (the value can be further configured in the server settings `appsettings.json, section TokenManagement -> rememberMeRefreshExpiration`).

Note:
While RayVentory Data Hub does not save your password directly, it is still not recommend to use this checkbox, especially when signing-in from a shared or untrusted machine.

The login screen let you also change the UI language, which will be remembered for the session. In order to change the language, press the language caption (for example **EN**) and select the desired one. In this version, both English and German are available.

Working with Multiple Tenants

After the default installation, a default tenant is installed and pre-configured. Later on, more tenants can be added to add a separation layer between different projects, customers or other purposes. A user can have access to one or more tenants. The selection of the tenant happens initially after signing in. If your current user has access to more than one tenant, a selection dialog will be shown after a successful login (but before going to the Home screen).

Click on the tile representing the required tenant/project and confirm by pressing **LOGIN** to jump to the Home screen.

Chapter The header shows how to change the current tenant after the user is signed-in.

Default Users

Each new installation of RayVentory Data Hub has the following default users:

User type	Name	Password	Description
Root	`root`	`raynet`	The root user has full access to the whole

User type	Name	Password	Description
			instance, can activate it, and manage the projects. When configuring the product to work in multi-client scenario, the root user must be used to define the clients.
System	System	-	Internal System-User that is only used by the Server and Data Hub Agent to reflect its actions, e.g. Agent sign up. Under no circumstances should this user be used manually

The initial log-in to RayVentory Data Hub should be performed with the root user. This user is elevated and allowed to perform all global tasks, including the setup of tenants, adding or changing the product license, and define global users. The root user is not meant for production. Configure at least two tiers of user levels (administrators and users) by creating individual users and assigning correct permissions to them. It is also possible to use predefined groups to simplify the latter task.

WARNING
The root user exists on every freshly installed instance and always has the same password (as listed in the table). It is important to change the password to a unique, non-trivial password before moving the instance to a production environment. Do **not** expose your instance to public access until the root password is not changed to a strong and secure password. **Failing to comply with this rule creates a serious security risk!**

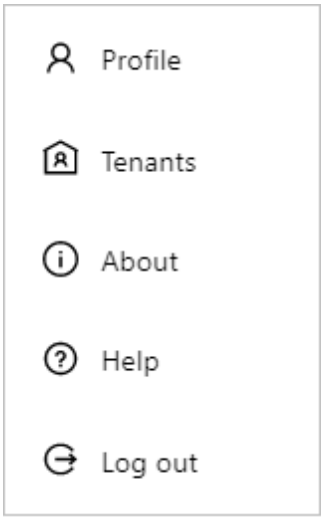
Basic Areas

These chapter describes some basic navigation areas, which are constantly visible regardless of the current screen or window.

The Header

The header is always visible. To the far right are the profile picture, name and role of the current authenticated user.

In the top-right corner the current username and the role are displayed. Hovering over the profile information at the top right corner opens a submenu with the following options:

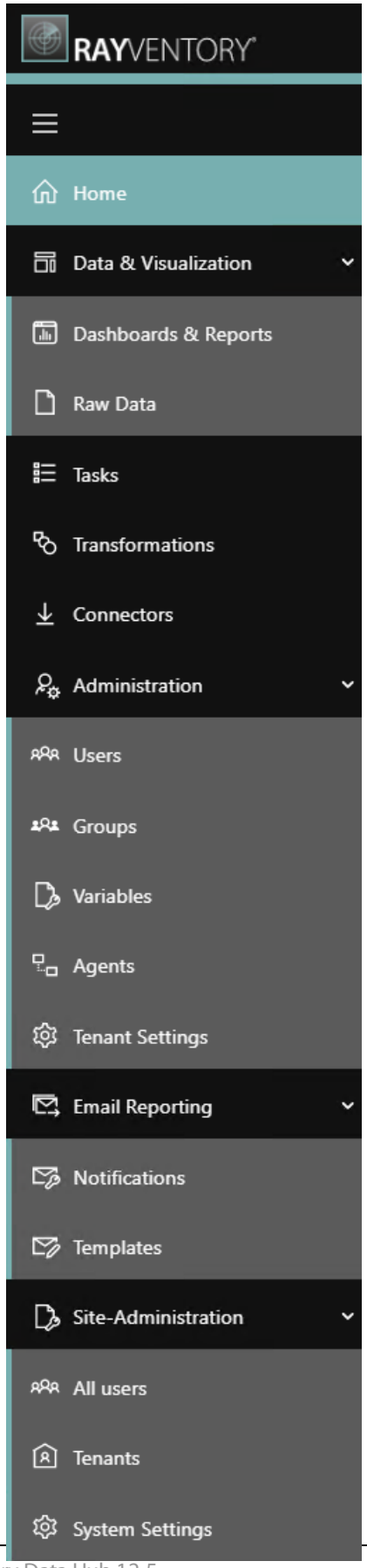


- **Profile**
  Navigates to the profile page of the current user
- **About**
  Navigates to the about page that provides general product information of RayVentory Data Hub
- **Tenants**
  Opens a dialog, where the current tenant/project can be changed
- **Help**
  Opens a PDF version of this user guide.
- **Log out**
  Logs the current user out and redirects to the login page

Product UI language can be changed by pressing the language caption in the corner. In this version, English and German are supported.

The Navigation

The navigation panel on the left border is always visible. It can be collapsed and expanded using the hamburger button at the very top. By selecting a menu entry the respective view is navigated to. The example image shows all menu entries available.


RAYVENTORY®
Home
Data & Visualization
Dashboards & Reports
Raw Data
Tasks
Transformations
Connectors
Administration
Users
Groups
Variables
Agents
Tenant Settings
Email Reporting
Notifications
Templates
Site-Administration
All users
Tenants
System Settings

Note:
Some menu items may not be visible, depending on the current permission level resulting from the user's role.

Reporting Objects

RayVentory Data Hub provides two basic types of reporting objects:

- Dashboards
- Reports

Dashboards are highly interactive, customizable, web-oriented controls, which aggregate the data and visualize them in a touch-friendly way. A dashboard usually takes the full width and height of the current view port. The current view port is not scrollable, which means that all controls, tables and other widgets must fit. More data can be shown in tabs, or in separate containers, where local scrolling is enabled.

Depending on the dashboard configuration, some advanced scenario like master-child patterns, filtering, selection etc. are supported. Most of the dashboards provide at least some basic level of interactivity, supporting mouse clicks, drag-and-drop, touch etc.

The primary goal of a dashboard is to visualize the data on a computer monitor or on a hand-held device screen. They are usually less suited for printing, where Reports are a better choice.

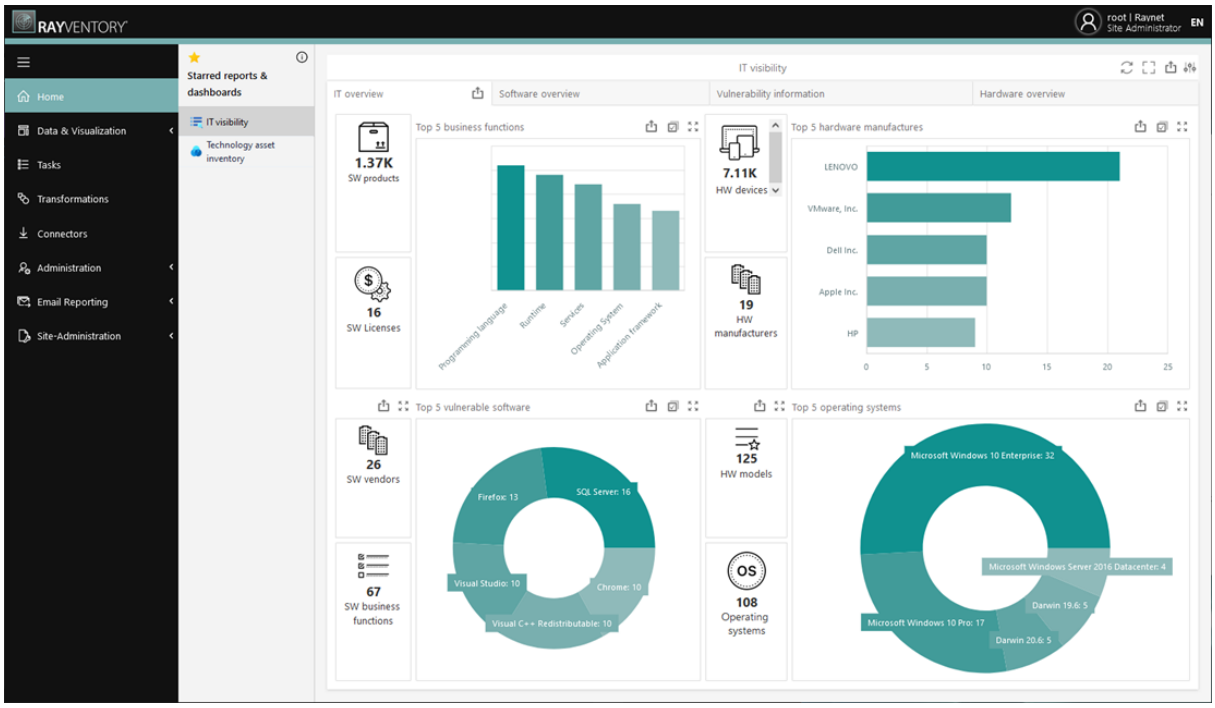
Reports are the static counterpart of dashboards. They are usually meant to be printed (physically or to a PDF/XPS format) and most of them consist of large tabular data and charts. Since the reports have been optimized and are well-suited for documents resembling sheets of paper (for example in A4 format) they contain many options for headers, pages, and fixed page elements.

Some limited interactivity is still possible, for example a report may support customizing the data sets and/or data shaping via input parameters. These - when supported - are shown in the web UI and usually have a form of a dropdown or text.

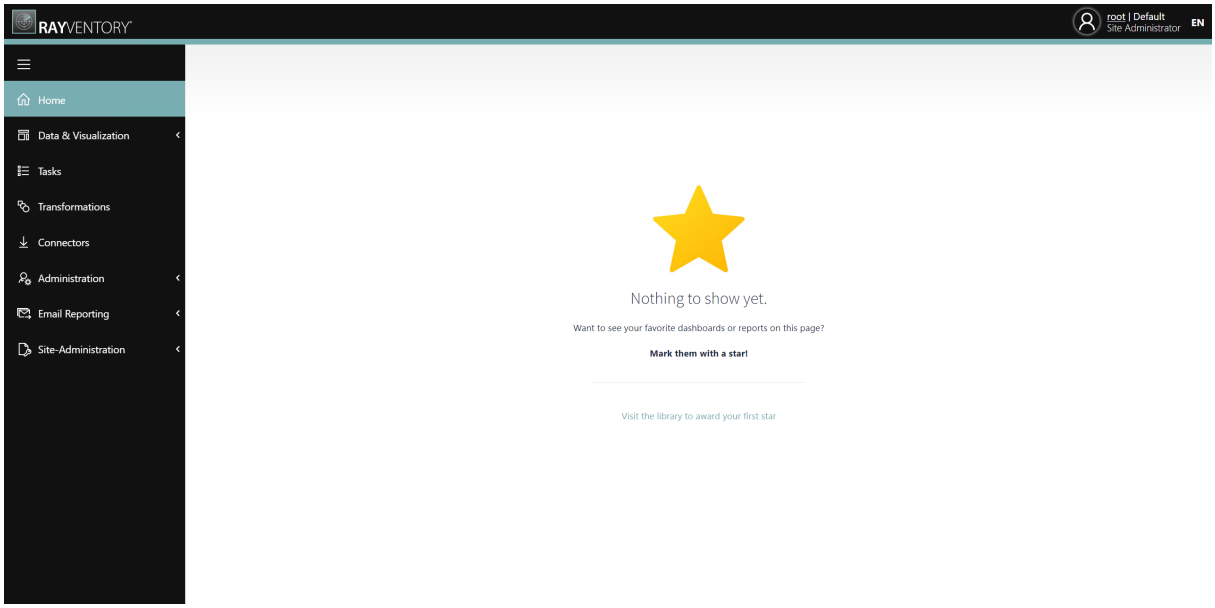
Additionally, there is also a special object called **Section** which functions merely as a container or a group of reports/dashboard. Functionally, the meaning of a section is equivalent to the term "Folder" and should be understood as such.

Home

The **Home** view shows a list of all the reports and dashboards that have been marked as favorites and the report or dashboard that has been selected from this list.



If no reports or dashboards have been marked as favorite, the following screen will be shown.



Data & Visualization

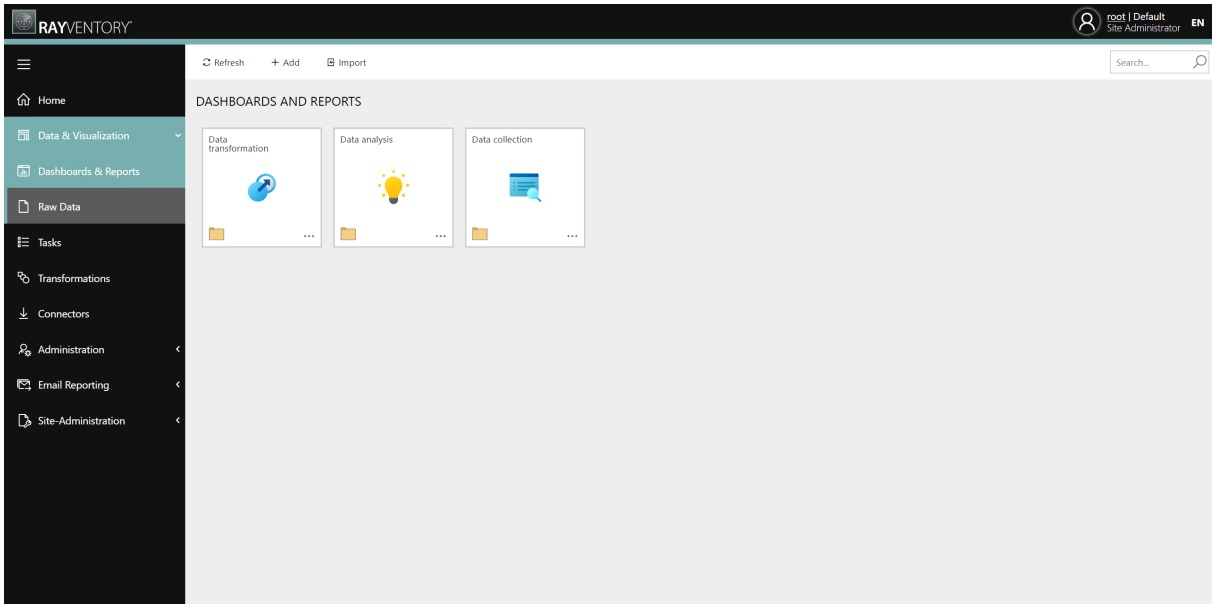
The Data & Visualization section contains all the data that has been collected as well as the reports and dashboards that refine the collected data. The section is divided into two subsections:

- Dashboards & Reports
  Contains all dashboard and reports that are available in the RayVentory Data Hub tenant.

- Raw Data
  Contains the unrefined data that has been collected by RayVentory Data Hub.

Dashboards & Reports

The library provides a structured overview of all folders, reports and dashboards. Here the user can create new folders, reports and dashboards and edit existing ones. Folders are used for personal structuring and grouping of reports and dashboards.



In the top part of each screen, an action bar with function buttons and search input is available. Below the action bar, a breadcrumb with page title is shown - clicking on breadcrumb items triggers navigation between respective areas.
The following function buttons are available in the Library screen:

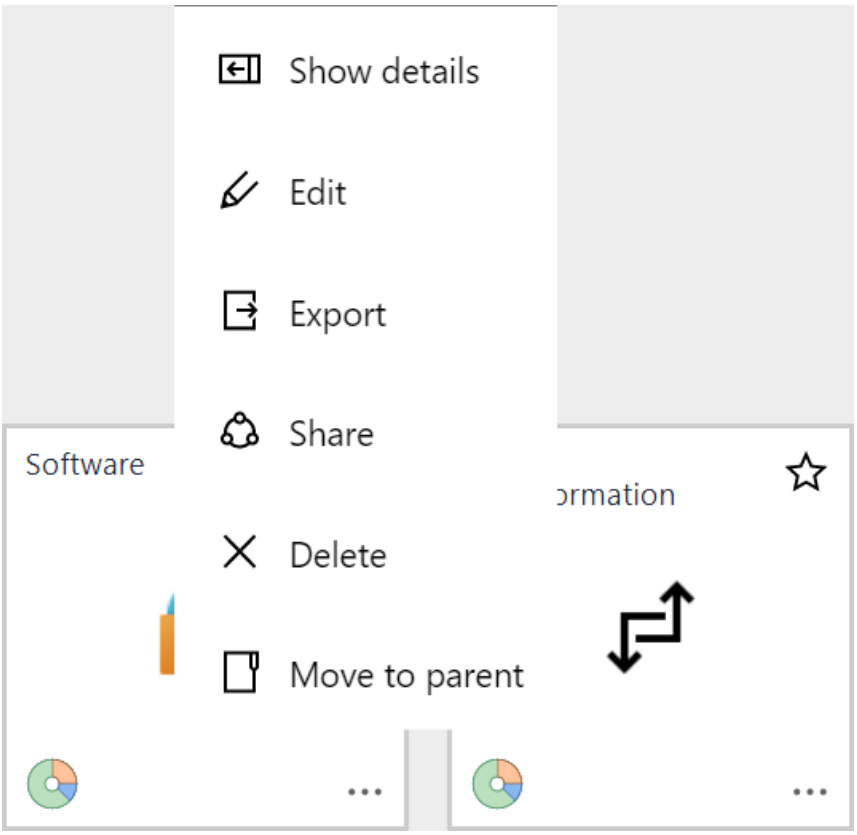
- **Refresh**
  Reloads all items of the current Section.

- **Add**
  Opens the Editor to create a new folder, report, or dashboard.

- **Import**
  Opens the import-editor that allows to import previously exported reports and dashboards. Please refer to the section Importing a Library Item for more details.

- **Search bar**
  Global search of sections, reports and dashboards by name.

Each folder, dashboard and report is represented as a tile, hereinafter referred to as Library Item. The name is displayed in the upper area and in the center the icon. Any logo in `.gif`, `.jpg`, `.jpeg`, or `.png` format with a maximum file size of 400 KB can be used.



Three dots in the the lower right corner represent a drop down menu. This menu has five actions:

- **Show details**
  Navigates to the detail view of the library item.

- **Edit**
  Navigates to the detail view and opens the editing panel.

- **Export**
  Navigates to the detail view and opens the export panel.

- **Share**

Navigates to the detail view of the library item and opens the sharing panel.

- **Delete**
  Deletes the library item. The user is prompt for confirmation before deletion.

- **Move to parent**
  Moves the item to the parent folder of the current folder.

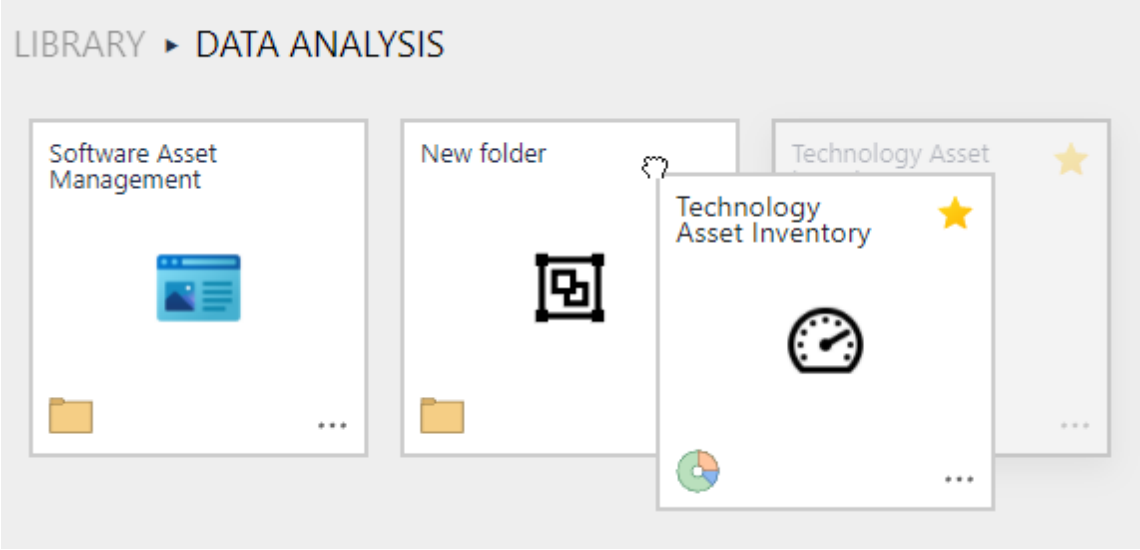
> **Note:**
> Menu options shown here might vary dependent of the authorization and permission level of the current user.

Clicking on the tile will have different outcomes dependent on the type of the library item:

- **Folder:** Drill down to the library items within this section.
- **Report:** Navigates to the **Report Viewer**.
- **Dashboard:** Navigates to the **Dashboard Viewer**.

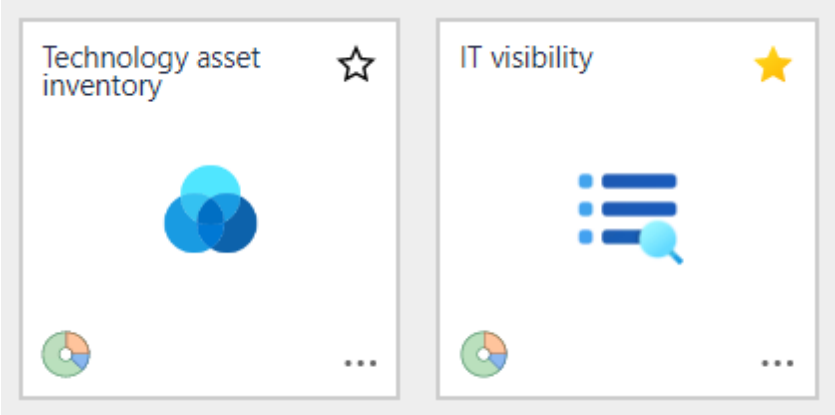
Reordering Items

The items can be reordered using drag-and-drop.



Favoring a Report or Dashboard

To add a report or a dashboard to the list of favorite reports and dashboards click on the star in the tile. If there is a white star with a black frame, it is not a favorite. If there is a yellow star, the report or dashboard will be shown in the list.



In the example shown in the screenshot, the **IT visibility** dashboard is marked as a favorite while the **Technology asset inventory** dashboard is not.

Creating a New Item

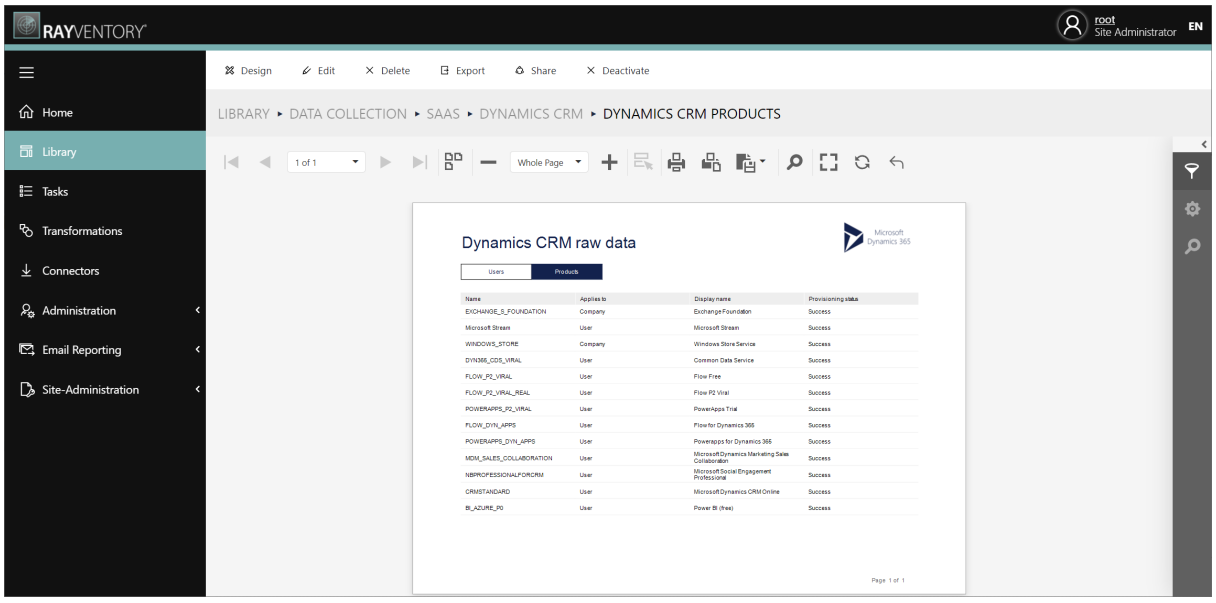
By clicking on the **Add** button from the actions bar in the library view opens a right side panel showing a form to create a new library item. Fill in the required fields:

- **Type** – The library item type (Folder, Report or Dashboard)
- **Name** – Name of the library item
- **Icon** – (Optional) Upload an icon, which is displayed in the center of the tile

Confirm the inputs by clicking on the button **Save changes** at the bottom. If the user is successfully created, a green toast notification is displayed in the top right corner.

If any error occurs, e.g. not all mandatory fields were filled or a field requires a unique name, a red pop-up notification is shown with further information. Furthermore, any invalid input field is marked with a red border and displays a short error message below the input field.

Reports and Dashboard Details

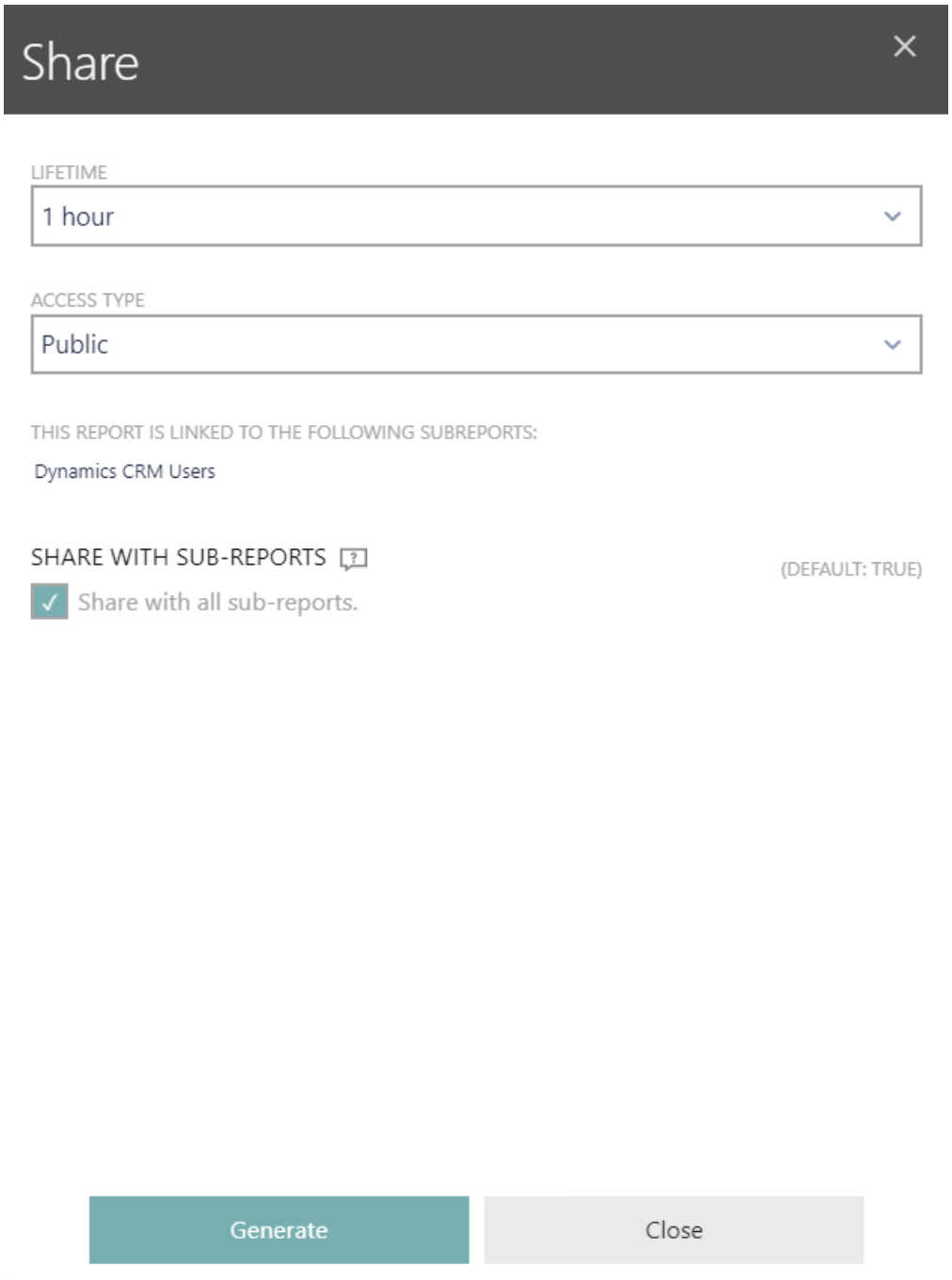


The action bar directly above the report has the following actions:

- **Design** - Switches the current view to the "designer" mode, in which it is possible to change, add or remove elements to the canvas.
- **Edit** - Opens the **Edit** panel that allows to edit the name, logo and mark the report as "featured".
- **Delete** - Deletes the current library item. This operation is irreversible.
- **Export** – Opens the **Export** panel showing a form to export the library item. Please refer to the section Exporting a Library Item for more details.
- **Share** - Opens the **Share** panel showing a form to create a sharable link of the library item.
- **Deactivate** – Deactivates the current item. This means that no user who does not have the Data admin or Administrator role can see or edit this library item, regardless of the permission set in the group the user is in. This button is only visible when the Library item is currently active.
- **Enable** – Activates the Library item again. This means that any user can see or edit this library item, dependent of the permission set in the group the user is in. This button is only visible when the Library item is currently deactivated

Note:
Action bar options shown here might vary dependent of the authorization and permission level of the current user.

Sharing a Library Item



By clicking on the **Share** button in the library item's drop down menu or in the actions bar at the library details view opens a right side panel showing a form to create a new shared link for the selected library item. The share option is only available for reports and dashboards.

By default, a public link with a lifetime of 1 hour is preselected in the form. Change the following settings as desired:

- **Lifetime** – Select a lifetime of 1 to 15 hours. After expiration of this time the link

becomes unusable

- **Access Type** – A link can be shared either public or non-public. A public link can be viewed by any anonymous user. A non-public link can only be viewed by authenticated users

Note:
A shared non-public link can be viewed by any authenticated user regardless its user role and permission. However, each shared library item can only be viewed and never edited. Hence, a specified permission for the given library item has no influence on shared reports or dashboards.

There is also a list of all subreports which are linked to the report that is going to be shared. By checking the **Share with all sub-reports** checkbox, the listed subreports will also be shared together with the selected report.

Exporting and Importing

See the chapter Importing and exporting for an overview of import and export capabilities of report and dashboard definitions.

Viewer

When navigating to a dashboard through the library view the dashboard-tool is first shown in viewing mode. The **Document Viewer** displays a dashboard in viewing mode. To enter the Dashboard Designer, press the **Design** Tab in the upper left corner above the dashboard's name.

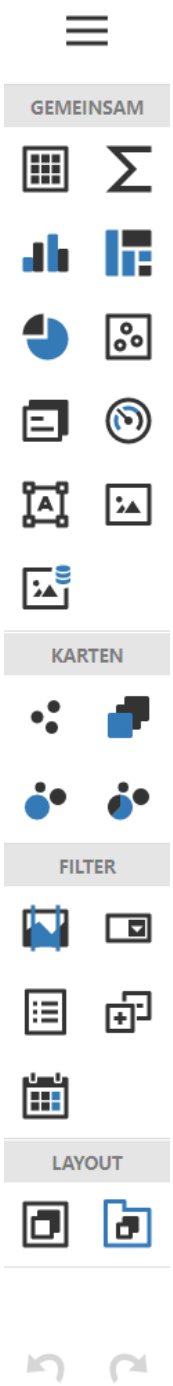
This viewer provides two buttons in the top right corner related to dashboard viewing and exporting:

- **Refresh** - Reloads the data.
- **Full screen** – Enters/leaves the dashboard's full screen mode
- **Export to** – Allows the export to PDF, Image and Excel

More information about working with reports and dashboard can be found in the *RayVentory Data Hub Dashboard and Reports Customization Guide*.

Designer

Dependent of the user's permissions a designer view might be available. The Dashboard Designer allows the creation of data-bound dashboards and provides a rich set of charts to create custom layouts and simple way to organize and manage multiple charts.



The Design Surface displays the dashboard's structure and content with live data. You can use the tools from the designer's toolbox (left side panel) to design the dashboard. Dropping the controls from the toolbox into the design surface creates a dashboard item. Selecting a dashboard item displays additional menu options for configuration, data-binding, rearranging and resizing.

Detailed information about how to use the Designer can be found in the *RayVentory Data Hub Dashboard and Reports Customization Guide*.

Working with Folders

The reports can be grouped in folders. It is possible to nest many levels of items.

To create a new section, go to the **Library** page and press **Add**.

Select the type **Folder** and provide a name and (optionally) a logo. Both name and icon can be changed later by editing the folder.

Import Data from Template

After the installation, the default library of reports, dashboards and their respective tasks is empty. This makes it possible to start with a clean-cut, and to build the reports and required tasks from the bottom-to-top approach, with a full control and flexibility.

Certain reporting objects and tasks are typical enough, that setting them up from scratch may be undesirable, and a much more feasible solution is then to take an already predefined set of objects, optionally with smaller adjustments required to get the required data.

In RayVentory Data Hub, there are two dedicated functions that can help you out with a quick setup.

- **Preconfigured set of reporting**
  This is a recommended way to go for most of users. By following these guidelines, RayVentory Data Hub creates preconfigured set of reporting objects and tasks for typical tasks, like reporting from Active Directory, SCCM, RayManageSoft, RayVentory + integration of several other sources, including SaaS (JIRA, Microsoft 365) and catalog capabilities. You can then later strip down the reports or tasks that are not needed, or have them customized to better suit your needs. A list of the available reports can be found here: Appendix I: List of Default Reports and Dashboards.

- **Importing standalone reports from .rpa files**
  This option is particularly useful for migrations and moving the data between two RayVentory Data Hub instances. More information about this route can be found in chapter Importing and exporting.

WARNING
If any of the default reports and dashboards have been customized (no matter how marginal the change is), it is of crucial importance to create a backup of the report before importing reports and dashboards from the template! If they are part of the import any customized reports might be overwritten by their imported counterpart!
It is recommended to always backup all reports and dashboards before importing the default reports and dashboards!
Backups can be easily created by clicking on the **BACKUP CURRENT DASHBOARDS AND REPORTS** button which can be found under **Administration** > **Tenant Settings**.

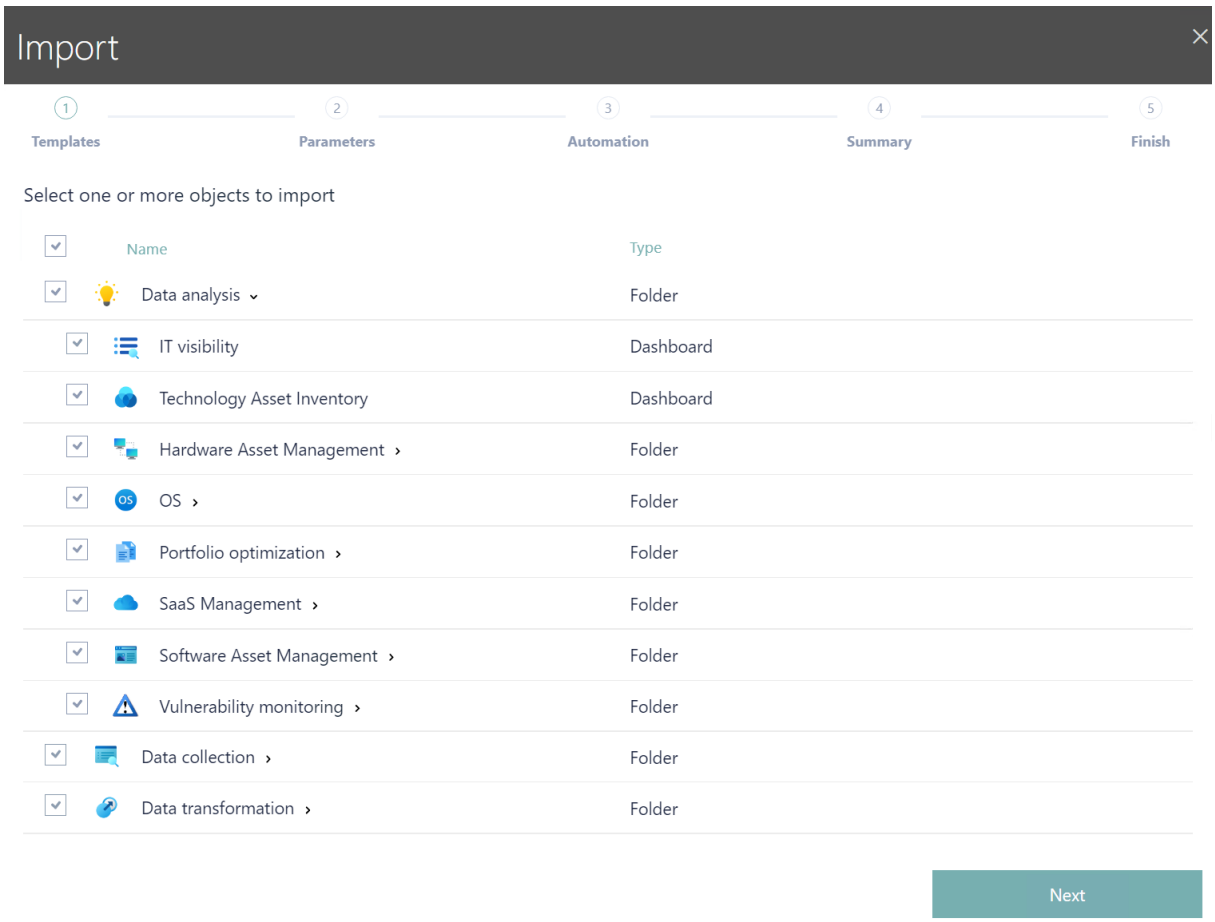
Creating Reports from the Wizard

RayVentory Data Hub bundles several reports, which can be imported in a wizard-like dialog.

To start the import process, go to the **Library** page and from the top menu select **Import** > **From template...**.

Note:
The wizard can be restarted as many times as needed, each time adding the required content. After a fresh installation and if your library of reports is still empty a small popup will be shown asking you whether you would like to start the wizard.

The first page of the wizard shows a selection of available templates for various dashboards and reports. Use checkboxes to select the required content. You can press the little right arrow next to each item to show its subcontent:



Once the selection is ready, press **Next** to go to the next configuration screen.

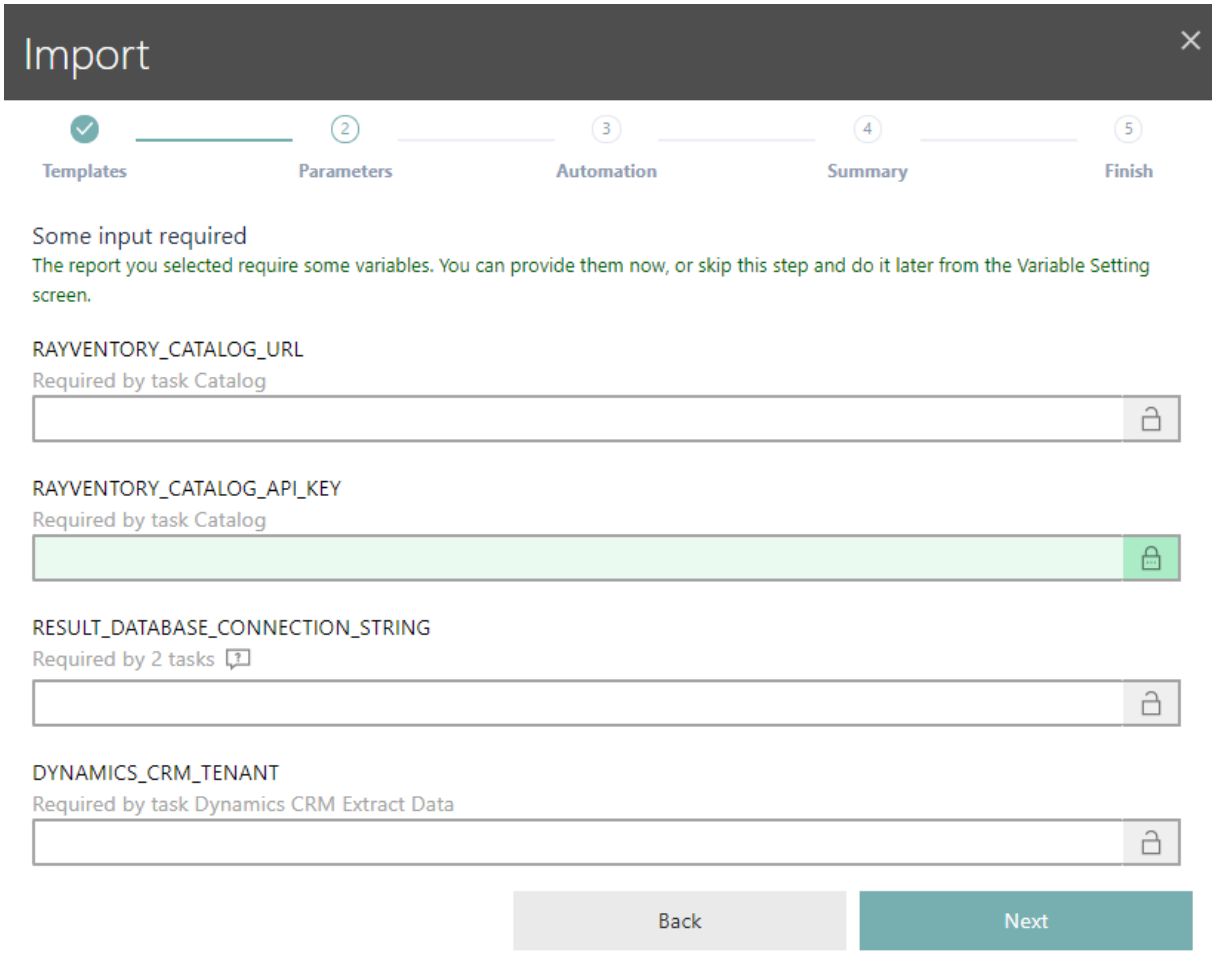
Note:
You may select one or more items to import. Selecting the whole library of templates is also possible. Bear in mind, that the import can be also split into smaller steps, as you can re-run the wizard as many times as needed.

Each reporting object defines a set of tasks, describing how to collect and extract the data from external sources (on-premise, cloud) to display them. Predefined dashboards and reports are pre-configured to use values and options that make sense for most enterprises. Certain aspects may require additional configuration and input from the user - for example logins and passwords, connection strings, URLs and other values. Some of them may be critical for respective task to successfully run - for example, a task that collects the data from a Microsoft SQL Server database requires a connection string, and without it it has no way to find out where the source of the data is.

Most of parameters can be customized later at design-time, by editing reports, dashboards or tasks. The parameters that are considered critical are collected by the wizard in form of "Variables".

Variable is a pair of identifying key and respective value, which is stored outside of the main reporting object, and available as a reference in other tasks. A variable has a unique name and stores a protected data that is separated from each tenant.

If the reports selected on the previous page require the usage of variables, RayVentory Data Hub shows the overview of required inputs in form of a simple list:

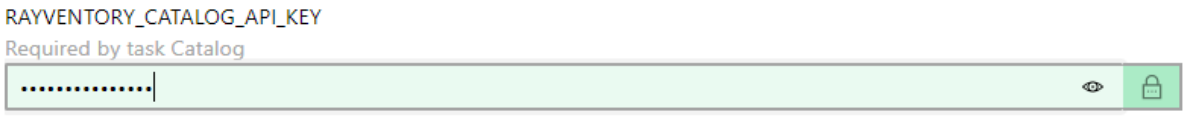


You can provide them now, or ignore for a time being and configure the values later on.

A variable may be clear-text or masked (protected). You can find more information about protection of variables in chapter Variables. A normal variable is a good choice for general-purpose data, which does not contain any sensitive information (login, password, token etc.).

An example of a normal (unprotected) variable, where the text is not masked:

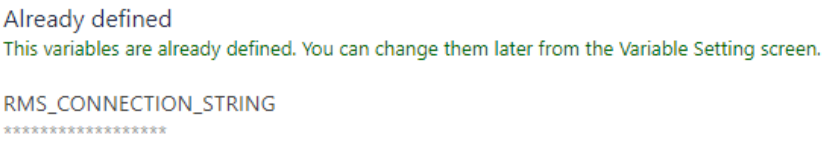
An example of a protected variable (with its content being masked):



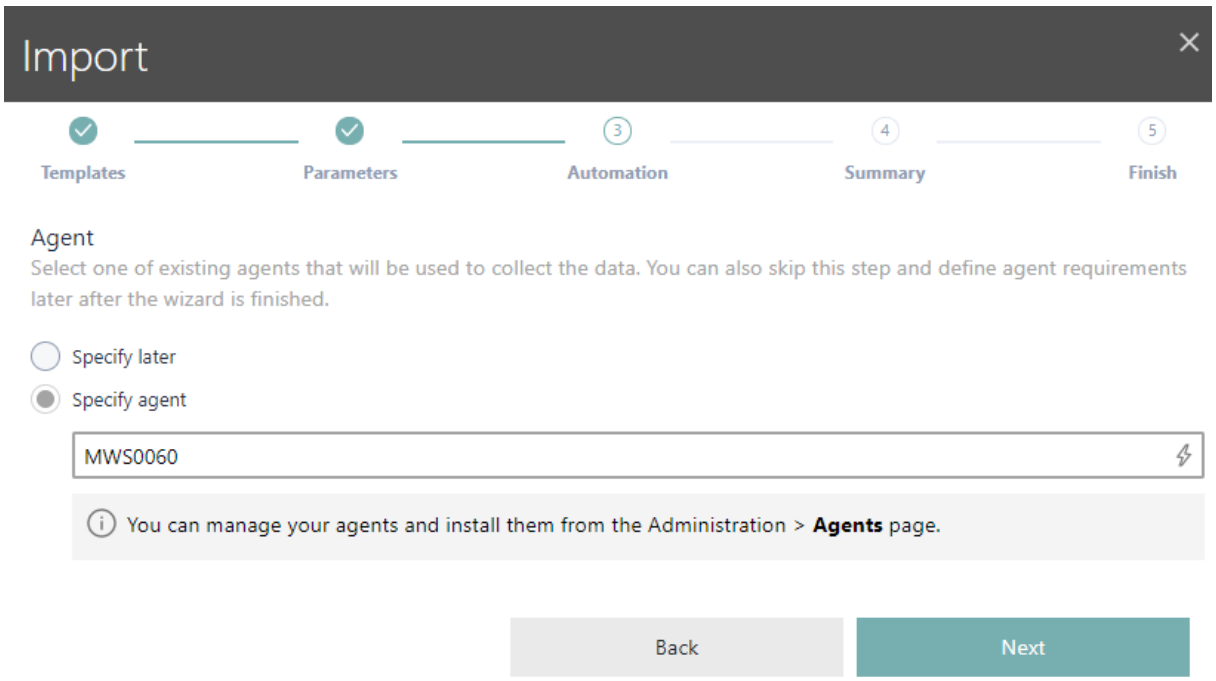
You can switch between protection levels by pressing the lock icon on the right side of the input. For protected values, you can also use the "eye" icon to temporarily show the edited content.

Note:
The choice of whether the variable is protected or not is totally up to you - the reports and tasks will behave the same. Internal encryption is transparent to the user, and there is no extra action involved for protected values. By default, RayVentory Data Hub marks all fields containing a keyword like "key", "password", "token" etc. as protected, but you can redefine the default option set.

If a variable has been already defined (for example from a previous run), its value will be shown read-only. To edit the variable, go to the Variables page once the wizard is closed.

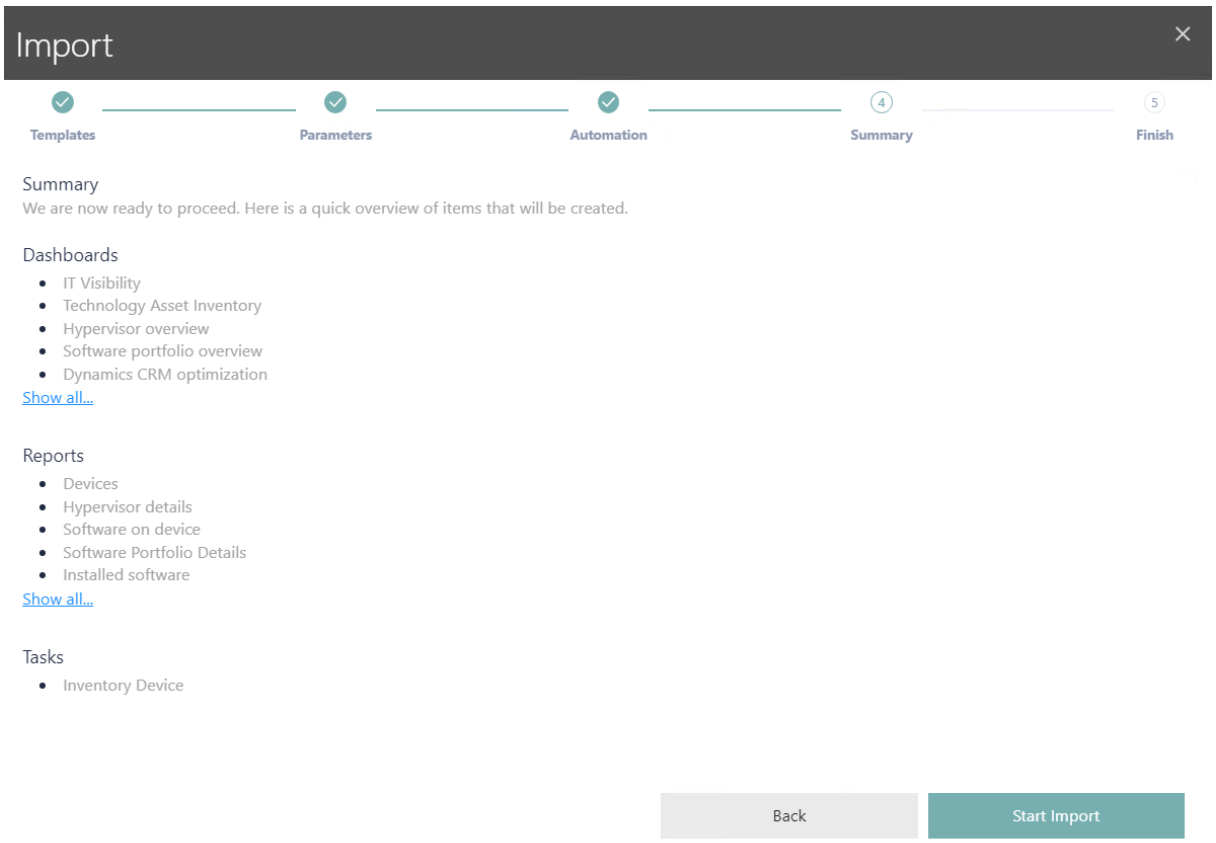


The imported reporting objects are static, that means they pull the data from database, but they are not responsible for collecting this data from external sources. This is a role of tasks, which are automatically created in the background. A task is a small unit of work, which targets a specific system and may optionally have a schedule. On the **Automation** page, it is possible to select which agent will be responsible for collecting the data. If you have no agents yet or the required agent has not been defined yet, you can ignore this step by selecting **Specify later** and pressing **Next**.

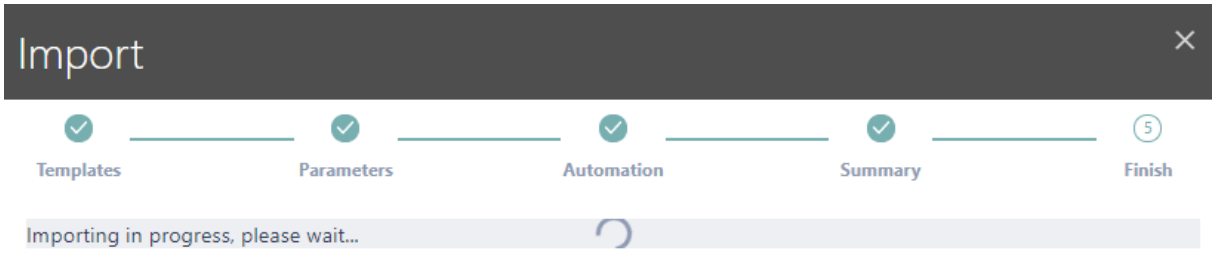


Otherwise, select the agent from the list. You can also go to the **Agents** subpage to manage your agents or install new ones. Once the required agent is selected, press **Next**.
More information about Agent management is available in the following chapter: Agents.

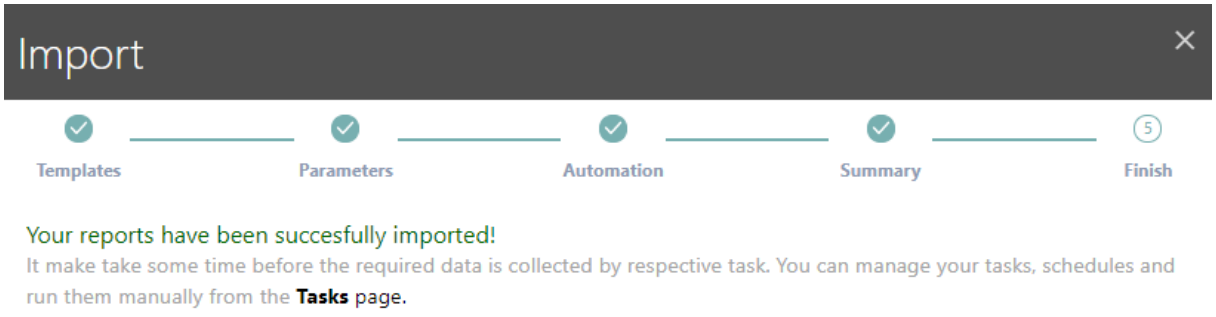
On the last page, the summary of selected objects, variables and automation options.



Importing can take a while.



Once the import is done, a confirmation message will be shown. You can close the wizard.



Note:
The reports will be initially empty. You need to start the respective task to get the data first. For more information, refer to the Tasks page.

Importing and Exporting

Importing to and exporting from the library is facilitated by dedicated functions.

Any dashboard or report can be exported directly from the UI. The result of export is a file with extension `*.rpa`. This file contains the following information:

- The definition (data source and visuals) of exported reports and/or dashboards,
- The definition of tasks required to gather the data,
- Linked reports,
- Further meta data information required to gather the data.

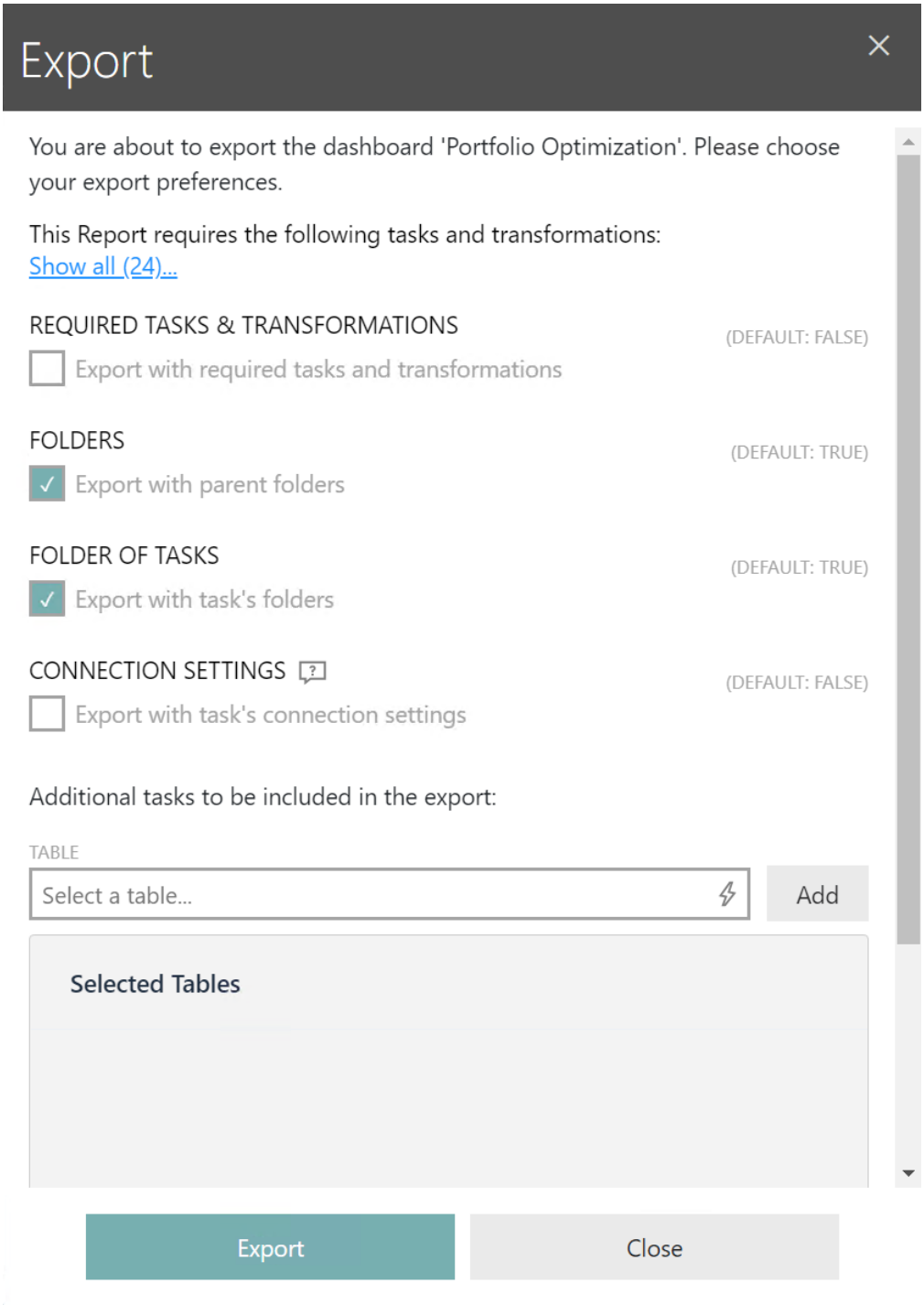
The exported file is self-contained, and is sufficient to move reporting objects and tasks between different instances, or for backup and migration purposes.

Note:
Depending on the options selected during the export, there may be some sensitive data (passwords, connection strings etc.) being exported in RPA files. When exporting your reports, make sure to either uncheck the option exporting the connection settings, or replace them with variables (`###VariableName###` syntax) by referencing the data from **variables**.

Exporting Reports and Dashboards

In order to export a report or a dashboard:

1. Navigate to the **Library** screen
2. Locate the object to be exported, and press the three dot menu (**...**)
3. From the context menu, select **Export**
   a. Alternatively, you can also click the report logo to open it, and then press **Export** in the top toolbar
4. An export dialog will be shown:



5. The reports are detached from the data, which is not going to be exported. Since you may be interested in recreating some of them on the target machine, it is important to define which data tables are to be considered, and consequently which tasks will be bundled with your report.
   a. Enter the names of the table to be exported. When in doubt, you can consult the settings from the Designer screen.
   b. RayVentory Data Hub will use the information provided in the previous step to determine which tasks are affected and must be exported as well. This happens automatically in the

background.

c. You can also configure whether to include the following additional information:
   i. **Required tasks**: The tasks to collect the data will be included in the exported content, and will be imported on the target system when the file is imported
   ii. **Folder of tasks**: This will preserve the structure of folders for affected tasks
   iii. **Connection settings**: This will include the details from the **Connection** tab of exported tasks. Please note that this option should only be used if the connection details are public or use variables - the configuration is stored clear-text in the exported RPA files, so anyone possessing the file can access the properties

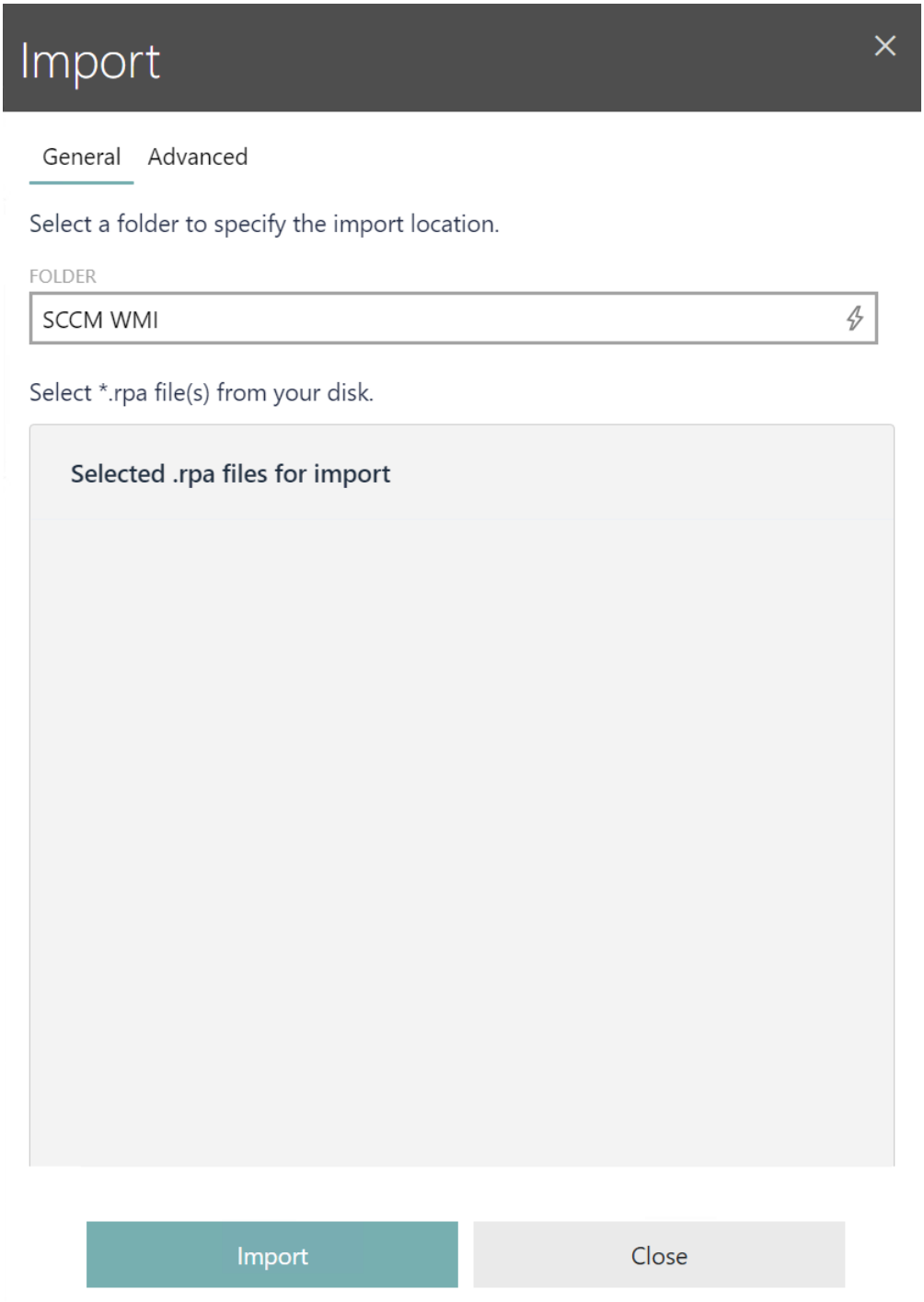
6. Once ready, press **Export**. The exporting will be started on the server, and once the results are available you will see an usual file prompt, asking about where to save the exported `.rpa` file

Note:
The export does not include the data stored in the Result-database bound to the report or dashboard. On the target system, you have to either re-run the imported tasks, or import the database manually (for example with Microsoft SQL Server Management Studio).

Importing Reports and Dashboards

In order to import a report or a dashboard:

1. Navigate to the **Library** screen
2. If you plan to import new objects to a not-yet-existing section, make sure to create it (press **Add** and create a new section)
3. In the toolbar, press the **Import** button
4. The **Import** dialog will be shown:

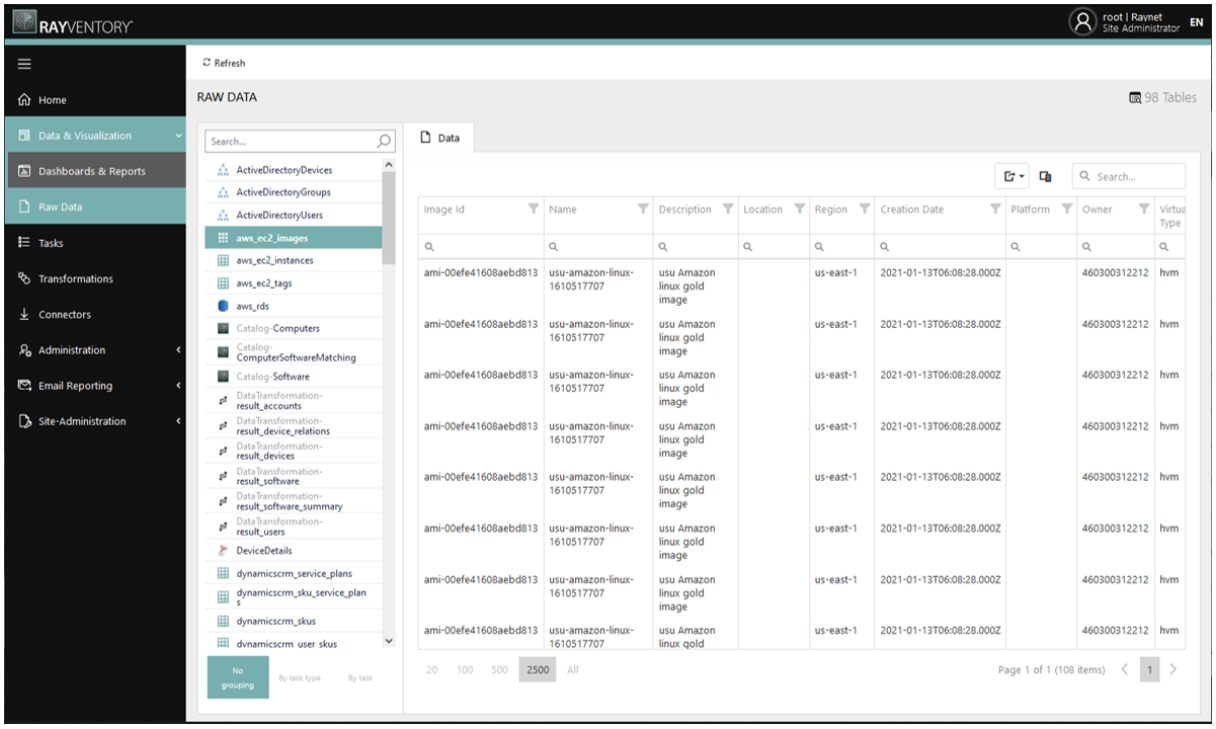


5. Select the target folder. If not specified, the root folder will be used instead
6. Select the required file (or more files at once if required)
7. In the **Advanced** section you can specify whether to skip or overwrite tasks in case of conflicts, or whether to import the parent sections which are defined in the RPA file
8. Once ready, press **Import** to start the importing. After a moment, a confirmation will be shown, informing about the imported content

Note:
The import does not include the data that was bound to the report or dashboard. Only the Tasks that create the required tables are imported. Thus, the user himself must ensure that the imported tasks are executed at least once after importing. Only then can the imported report or dashboard display the actual live data.

Raw Data

The Raw Data section contains the unrefined data that has been collected by RayVentory Data Hub and saved to the database.



The section is divided into two different areas. The first area contains the table list. If a table has been selected in the table list, the second area shows the contents of this table. At the top right of the screen the total number of tables contained in the database is shown.

Table List

In the table list, all tables that are available in the database are shown. In order to find a specific table, it is possible to use the search field on top of the list to lower the number of tables that will be shown.

Furthermore, there are three grouping options that can be used for a better overview at the bottom of the list. To activate one of these grouping options, click on the specific field. The following options are available:

- No Grouping
- By Task Type
- By Task

No Grouping

If this option is selected, an alphabetical list of tables will be shown, similar like in the SQL Server Management Studio (SSMS). Additionally, an icon taken from the task that created the table or a

generic icon if the task is unknown, will be shown next to the table.

By Task Type

If this option is chosen, the tables will be grouped by task type and a header for each of the groups will be shown. The tables themselves will than be shown with generic icons, as the task type icon will already be shown in the header. Example task types would be Microsoft Server, Active Directory, Transformation, etc.).

By Task

If this option is chosen, the tables will be grouped by task and a header for each of the groups will be shown. The tables themselves will than be shown with generic icons, as the tasks icon will already be shown in the header. An example for a task would be **Get Active Directory Groups**. A list of the default tasks can be found here.

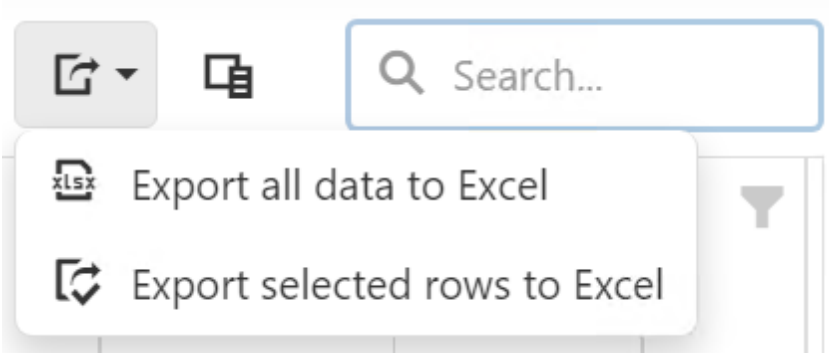
Data Section

In the Data section, the content of the selected table is shown. The content can be exported and sorted by multiple means. Specific information can be searched using the Search field and it is also possible to filter the different columns using the filter icon located at the top of each column next to its name. Furthermore the following two options are available as buttons on the top right side of the section:

- Export
- Column Chooser

Export

The **Export** button can be used to export the data of a table to an excel file. There are two export options available.



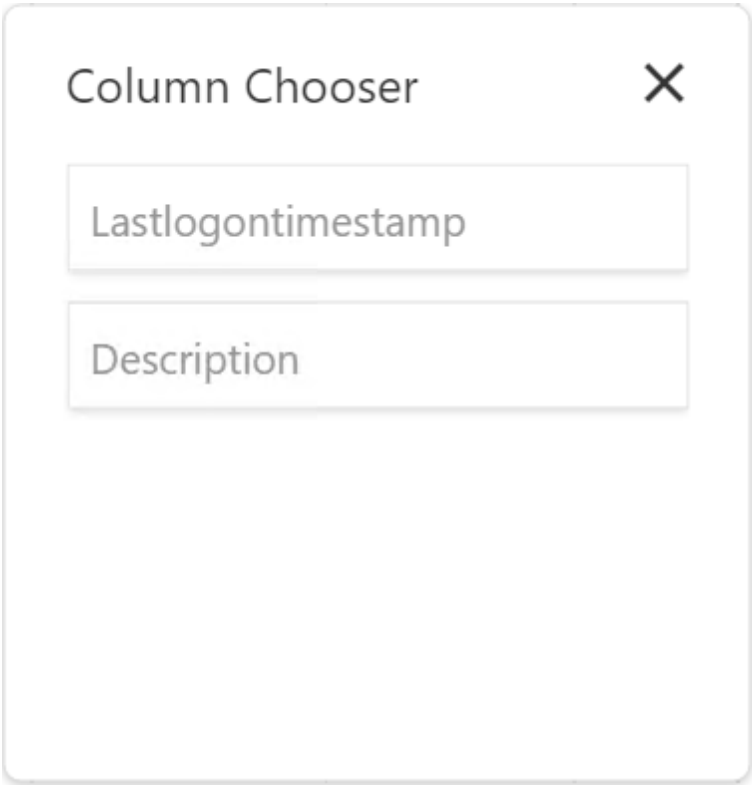
- **Export all data to Excel**: This option can be used to export the whole table to an excel file.
- **Export selected rows to Excel**: This option can be used to export the currently selected rows to an Excel file.

The files will be exported as `DataGrid.xlsx`.

Column Chooser

To open the Column Chooser, click on the Column Choose button located in the upper right corner of the Data section.

By default, the Column Chooser will be empty as all columns of the selected table will be shown. Column can be moved between the data grid and the Column Chooser by using the drag and drop technique. To hide a column, click on it and drag it into the Column Chooser.

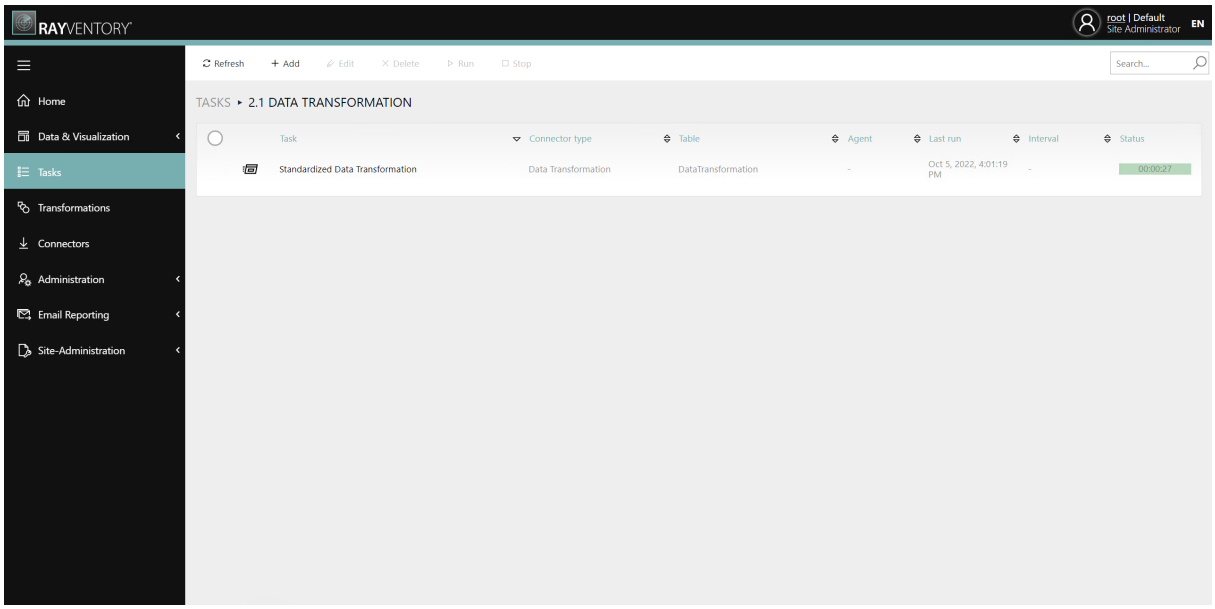


It will now be shown in the Column Chooser and hidden from the data grid in the Data section. To show the column again, click on the column in the Column Chooser and drag it to the data grid. It will be shown in the same location as before.

Tasks

Tasks are units of work that are executed on demand or via a scheduler to extract the data from external sources like databases, REST APIs, files, etc. The **Tasks** screen is a central management place for task-related activities, configuration, and triggers.

Note:
In order to access this screen it is necessary to be in the **Administrators** or **Data administrators** role.



The action bar contains the following function buttons:

- **Refresh**
  Reloads all tasks and updates the table.

- **Add**
  Opens the **Add Task** panel that allows it to create a new task.

- **Edit**
  Opens the **Edit Task** panel, which allows it to edit a single selected task. The Edit-button is only enabled when a single task is selected.

- **Delete**
  Deletes the selected tasks. A prompt will be shown to confirm the deletion.

- **Run**
  If the selected task can be started (its type is licensed, there is a valid agent configuration, and all settings are in place), this button starts it on demand, ignoring any schedule set on the task.

- **Stop**
  This option is active for tasks that are currently running. Pressing this button and confirming the action stops the task and rejects any results it collected so far.

Creating or Editing Tasks

Clicking on the **Add** button from the action bar in the tasks view opens a right side panel showing a form to create a new task. This panel consists of the folllowing tabs:

- General
- Configuration
- Connection
- Agent Settings
- Schedule
- Info

The availability of these tabs depends on the connector that is used for the task.

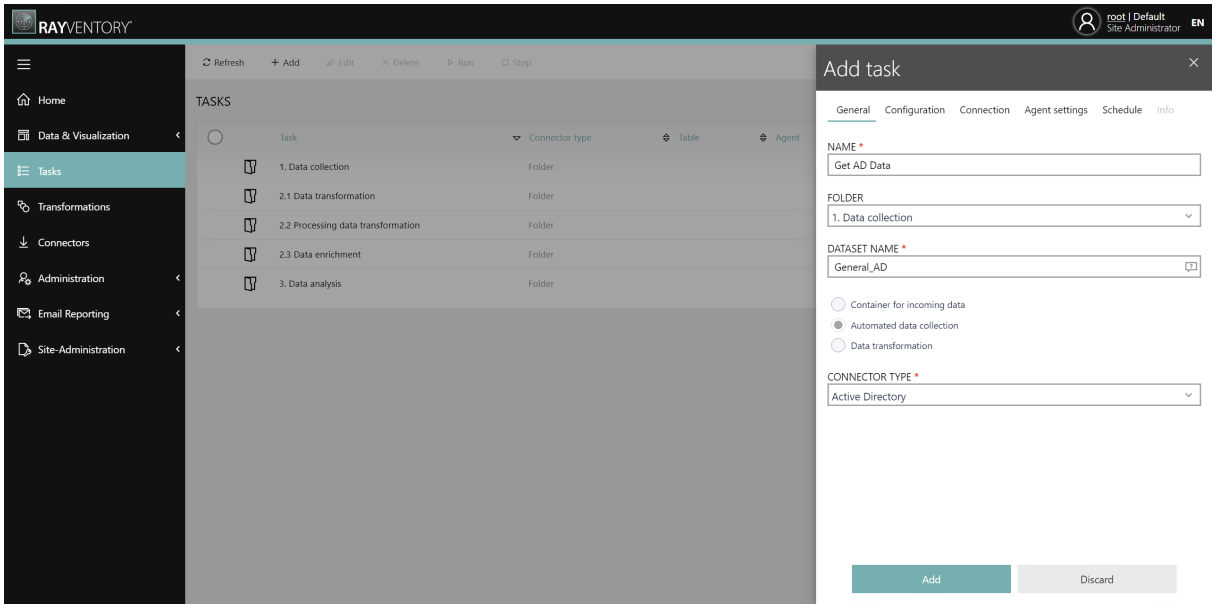
To edit a task first select it from the list and then click the **Edit** button.

To save the data, click on the button **Add** (when adding) or **Save changes** (when editing). Should any error occur, e.g. not all mandatory fields are filled, a red toast notification is shown with further information. Furthermore, any invalid input field is marked with a red border and displays a short error message below the input field.

A list of the custom tasks can be found here: Appendix II: List of Tasks for the Default Reports and Dashboards.

General

The **General** tab defines basic properties that identify the task and that influence other tabs, particularly the **Configuration** and **Connection**.



- **Name**
  The name of the task. It must be unique in tenant scope, and should describe what the task does.

- **Folder**
  If the task is meant to be in the root folder, leave this field empty. Otherwise select the parent folder in which the task will be saved.

- **Dataset name**
  This is the base name of the data set, where the data extracted by the task data collector will be saved. If the result of data extraction is a single table, then that table is going to be available under the name specified in this field. Otherwise, the set name will be used as a prefix, and the task data collector decides on his own what to append to the base name for a semantic and unique meaning. For example, the Active Directory data collector will write a few tables: if the data set name is *ActiveDirectory*, then the tables with results will be names *ActiveDirectory.Users*, *ActiveDirectory.Computers* etc. More information regarding the names of the tables can be found in the Naming Conventions chapter.

- **Container / Automated collection**
  There are three main ways how to provide the data. The most common use is to automate the data extraction by Data Hub scheduling or on-demand mechanisms, in which the whole configuration and heavy lifting is done by the Data Hub. Selecting the option **Automated data collection** activates several other tabs and fields. If you intend to create a task that functions as a container (without particular type assigned to it), select the first option **Container for incoming data**. In this case, tabs like **Configuration**, **Connection**, **Agent Settings** and **Schedule** will be hidden. This is a good choice for tasks that do not collect the data, but are rather a stand-in for automation via REST endpoints or manual upload of CSV files.

- **Container type**
  If the task is defined to collect the data, this required field is used to configure the built-in data collector that will extract the data. Select a type from the list. Bear in mind that not all

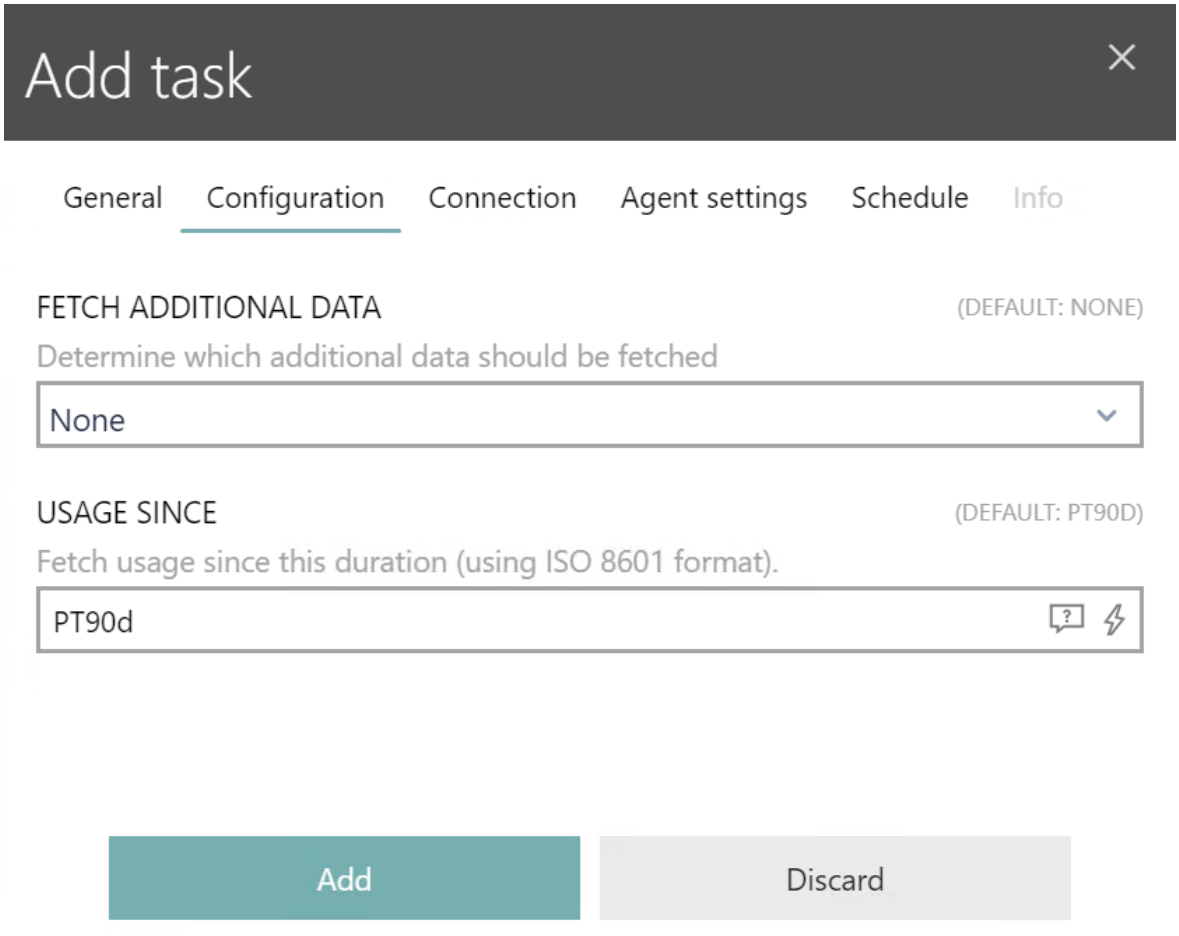
connector types may be available here - the actual choice depends on the configuration and licensing. More on that in chapter Connectors.

Note:
It is not possible to change the connector type of the automation type of an already existing task. In this case the connector type is a read-only field.

Configuration

This tab contains general task settings. Most of them are already set by default to reasonable values. The actual content of this tab depends on the current connector type. Below are some examples:

Every task requires other parameters and shows a dedicated UI. Below is an example of the configuration of Microsoft Dynamics CRM:



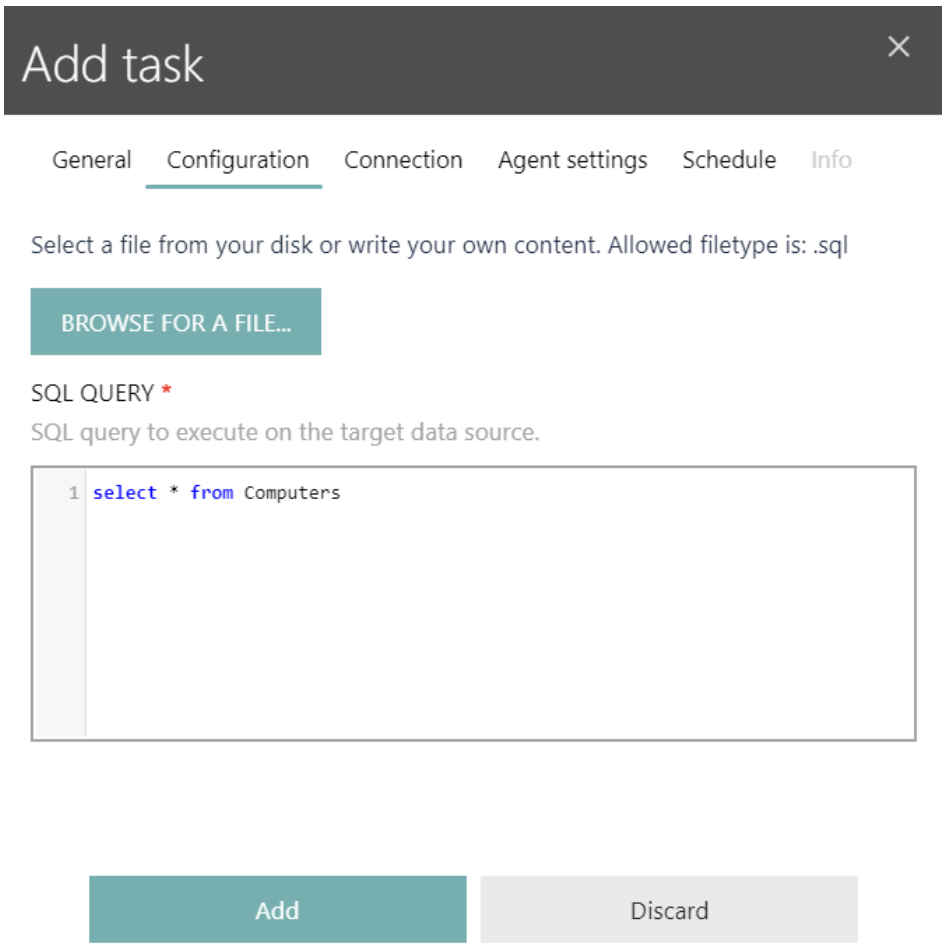
The fields marked with * are required. For optional fields you can leave the default value or enter a custom value as required per-task. As a rule of thumb, task configures what is to be done, and the parameters required for connection, authentication and authorization are defined in the **Connection** tab. If there is no configuration required / available, the whole **Configuration** tab may be grayed out.

The following chapters show how to configure specific common task types:

- Configuring Microsoft SQL Server Tasks
- Configuring Active Directory Tasks

Configuring Microsoft SQL Server Tasks

Microsoft SQL Server tasks require only the full query. You can use all syntax and language constructs accepted by the target SQL Server, against which the query will be executed by the agent. The query can be written manually into the editor, or uploaded from a local `.sql` file by using the **BROWSE FOR A FILE...** button.

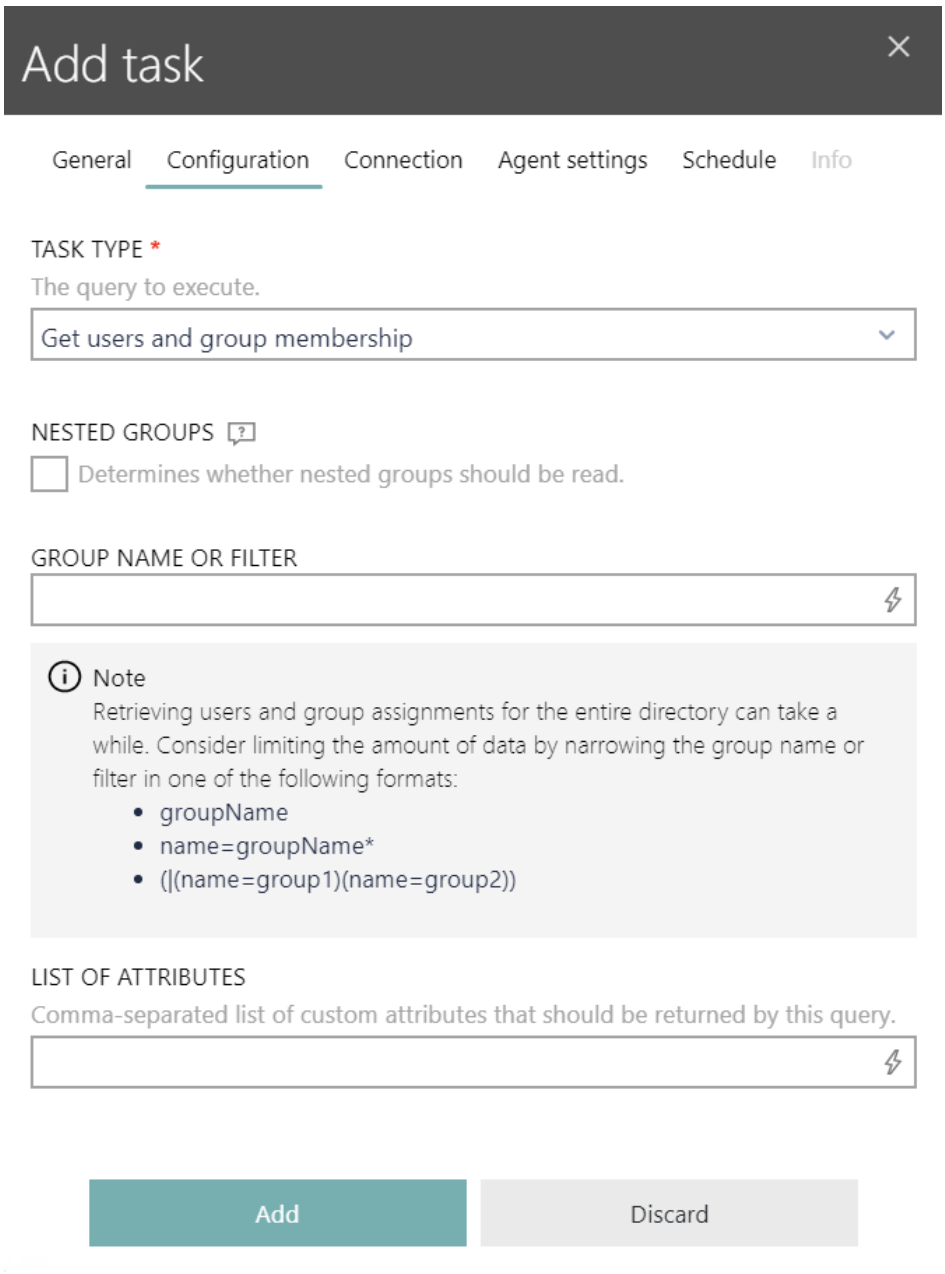


WARNING
RayVentory Data Hub uses a loose structure of the data and does not enforce any particular security and access rights in terms of what the reporting queries are doing. Since temporary tables are sometimes necessary and may be deleted afterwards, commands such as "delete", "drop", "update" are permitted. Thus, improper use can lead to data loss in the database source. Make sure that the query does not damage other reporting data by only removing or changing the tables owned by the report.

This only applies to the reporting data. RayVentory Data Hub uses a separate database to keep track of its settings, users, reports etc. which are separated and cannot be adjusted or damaged this way.

Configuring Active Directory Tasks

Active Directory tasks require the selection of one of multiple predefined types or a custom query. Additionally, several optional properties are available for configuration. In case of a custom selection, the complete filter query is to be specified by the user.



Configuring PowerShell Tasks

A detailed guide how to collect the data using PowerShell is described in the following chapter: Using PowerShell Connector.

Connection

The connection tab contains properties of two types:

- Technical parameters, for example time-outs or retry and back-off strategies,
- Authorization parameters, for example logins, passwords and connection strings.

Add task

General | Configuration | Connection | Agent settings | Schedule | Info

Authentication

TENANT *

The directory tenant that you want to request permission from. This can be in GUID or friendly name format.

CLIENT ID *

The Application ID that the Azure app registration portal assigned when you registered your app.

CLIENT SECRET *

The Application Secret that you generated for your app in the app registration portal.

DYNAMICS CRM URL *

The full URL (with protocol) to your Dynamics CRM instance. This URL is required to fetch usage from Dynamics CRM. When using native apps, make sure to configure application user for it (in CRM portal).

E.g. https://yourName.crm4.dynamics.com

Task Parameters

PAGE SIZE (DEFAULT: 100)

Set the page size (number of records) during a paginated API request.

100

USAGE REQUEST INTERVAL (DEFAULT: PT12H)

Duration of interval per request (using ISO 8601 format).

PT12h

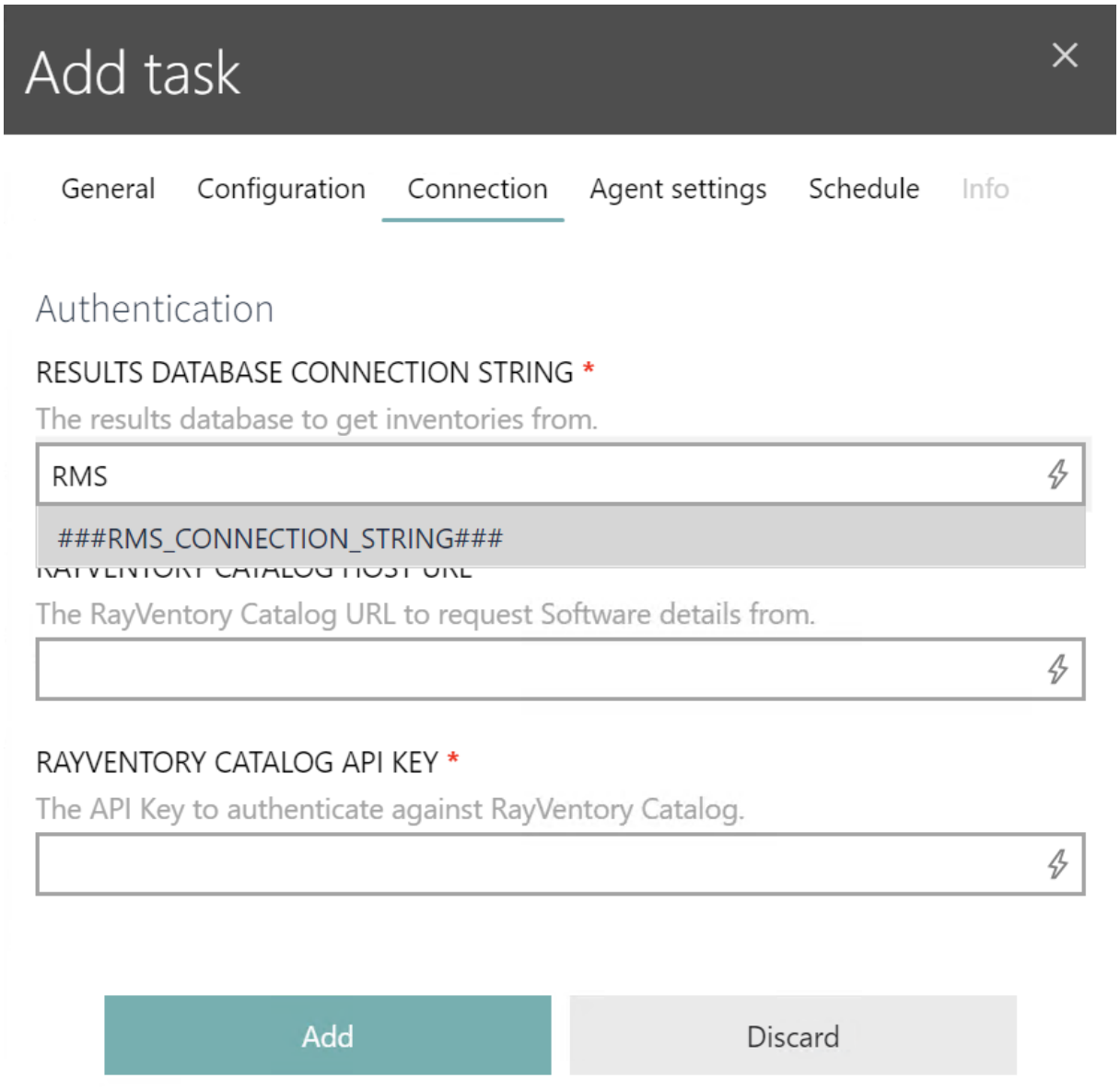
The fields marked with * are required. For optional fields it is possible to leave the default value or enter a custom value as required per task. If there is no configuration required / available, the whole **Connection** tab may be grayed out.

Referencing Variables

It is a common pattern to avoid hardcoded values in the **Connection** or **Configuration** tab. A good place to store them securely are the **Variables**.

To use a variable, first ensure that it has been already defined, by going to the Variables page.

All text fields that are marked with a "flash" icon support an intellisense input, in which typing a few characters offers suggestions from the Variables:



Once a proper variable is found, just click on it to include its name (together with opening and closing ###, being part of the syntax). The variable value will be resolved on runtime as the agent picks up the task.

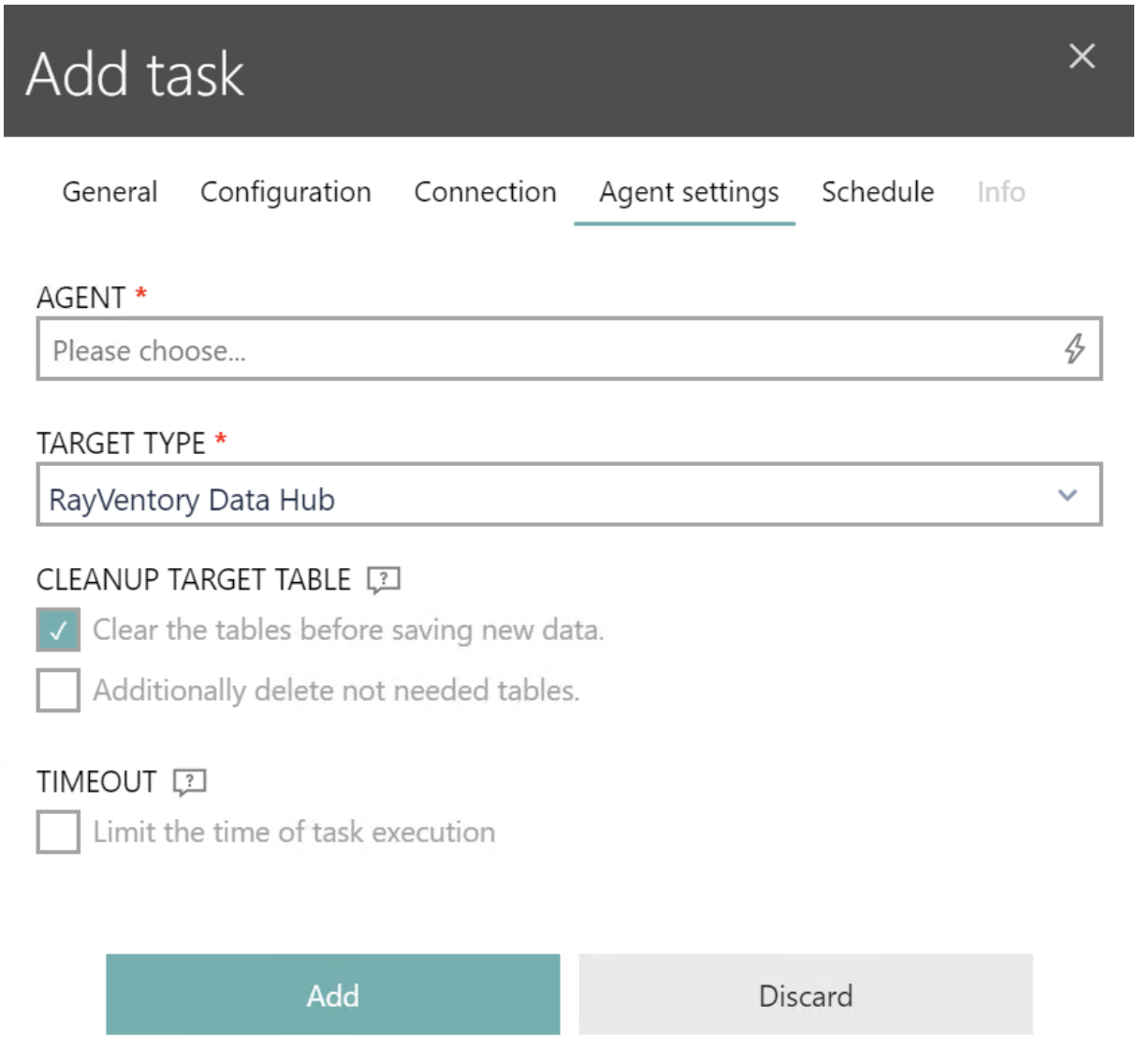
Using Variables for Password Fields

It is also possible to use variables for password fields. In this case, no intellisense dropdown is shown. Instead, type ###NAME### into a password box field, where NAME is the name of the variable. There will be no visual confirmation of whether the value of a password field uses a variable or not.

Note:
It is recommended to use only protected values for passwords, credentials and other sensitive information.

Agent Settings

This tab defines which agent picks up the tasks and how it reports back.



- **Agent**
  The agent selected from the list. This is a required field - all tasks marked for automated data collection must have an agent defined.

- **Target type**
  The target where to save the data. In most cases, using the default **RayVentory Data Hub** is the best choice. The other two options (SQL or ODBC) should be use in exceptional cases to transfer the data in complex environments, or to tweak the performance of large data sets.

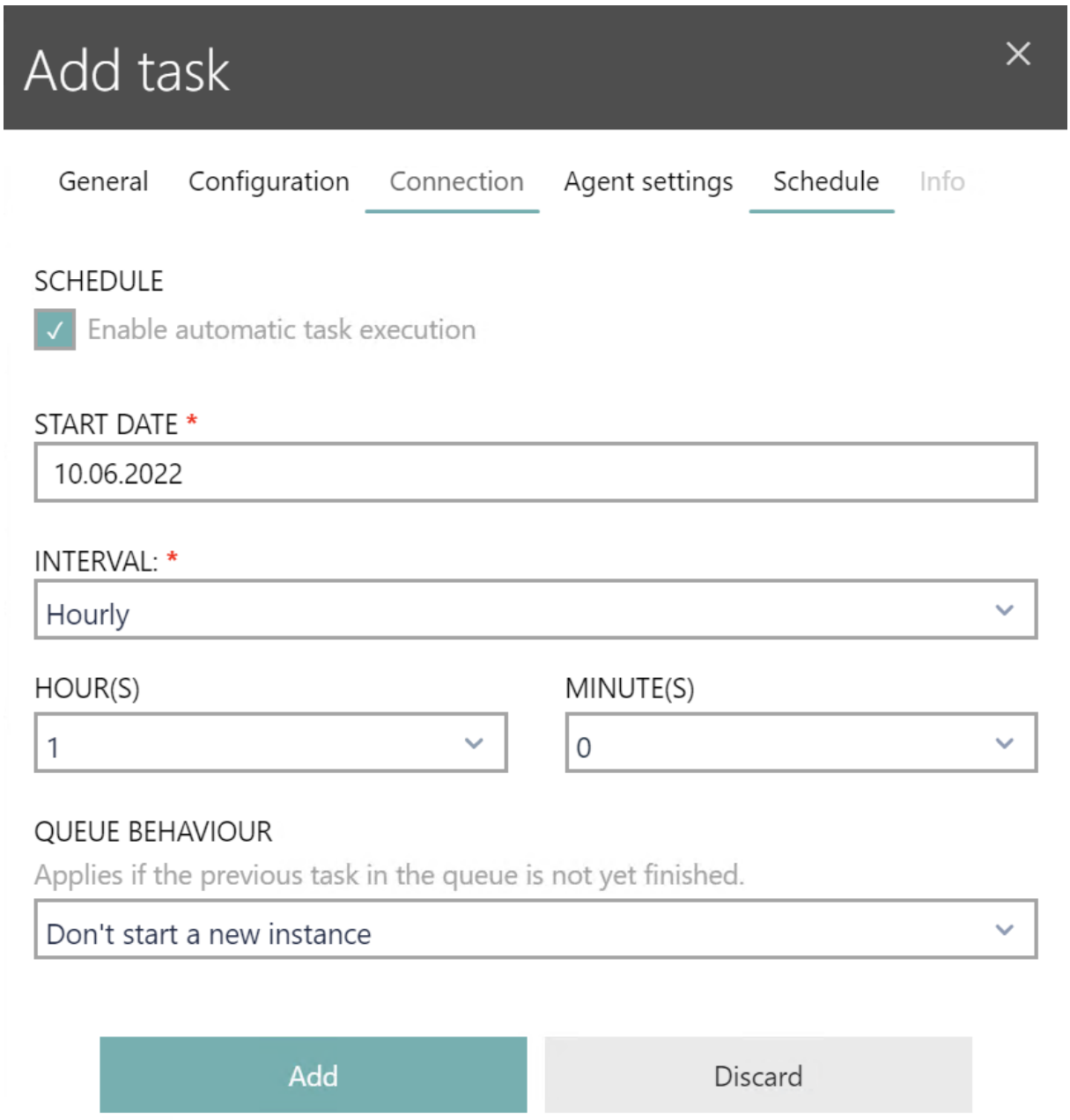
- **Cleanup target table**
  This setting defines whether to delete the content of the already existing table with the same name. If this option is checked, the previous data will be completely overwritten by the new

one. If you uncheck it, another checkbox will be shown, where the handling of duplicates must be defined (either ignore them or de-duplicate the data). Unchecking this option is considered an advanced feature.

- **Timeout**
  The maximum duration to wait for the task. Note that there may be different timeouts defined in the **Connection** or **Configuration** tab, but they are scoped locally and affect the execution on the Agent. On the other hand, this setting controls the whole process, including hand-shakes, data conversion and transfer and the actual extraction on the agent. If the checkbox is selected, a timeout in seconds must be provided.

Schedule

A task can be started manually, or on a schedule. To enable the schedule mode, select the checkbox **Enable automatic task execution**:



- **Start date**
  The initial date from which the schedule is active.

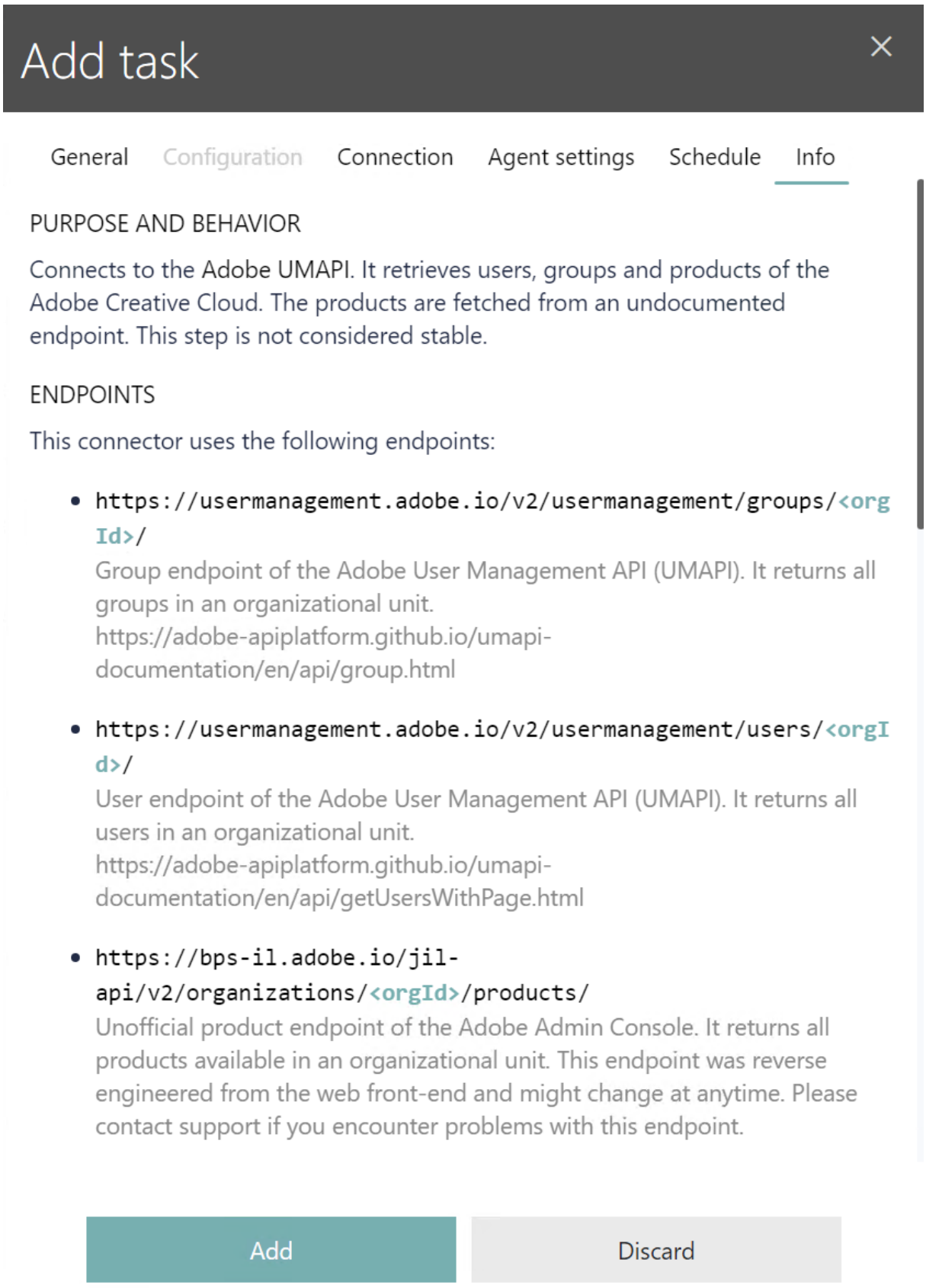
- **Interval**
  The schedule mode (minutes, hours, daily, weekly, monthly or advanced)

- **Interval options**
  Depending on the selected interval (previous drop-down) different options may be shown. If the **Advanced** interval is selected, a CRON expression is required as an input. Refer to internet sources, for example https://help.ubuntu.com/community/CronHowto to learn more about available options.

- **Queue behaviour**
  This configures what to do if two scheduled executions overlap. It is possible to either ignore the second (newer) execution, or have it run as soon as the current one finishes. The default option **Don't start a new instance** is usually the best choice.

Info

The Info tab contains all relevant information about the connector that is being used by the connector. This information is only available for cloud connectors.


Add task
General
Configuration
Connection
Agent settings
Schedule
Info
PURPOSE AND BEHAVIOR
Connects to the Adobe UMAPI. It retrieves users, groups and products of the Adobe Creative Cloud. The products are fetched from an undocumented endpoint. This step is not considered stable.
ENDPOINTS
This connector uses the following endpoints:
https://usermanagement.adobe.io/v2/usermanagement/groups/<orgId>/
Group endpoint of the Adobe User Management API (UMAPI). It returns all groups in an organizational unit.
https://adobe-apiplatform.github.io/umapi-documentation/en/api/group.html
https://usermanagement.adobe.io/v2/usermanagement/users/<orgId>/
User endpoint of the Adobe User Management API (UMAPI). It returns all users in an organizational unit.
https://adobe-apiplatform.github.io/umapi-documentation/en/api/getUsersWithPage.html
https://bps-il.adobe.io/jil-api/v2/organizations/<orgId>/products/
Unofficial product endpoint of the Adobe Admin Console. It returns all products available in an organizational unit. This endpoint was reverse engineered from the web front-end and might change at anytime. Please contact support if you encounter problems with this endpoint.
Add
Discard

Depending on the availability, the tab contains information about the purpose and the behavior of the connector, the endpoints that are used and the tables and columns that will be produced in the database.

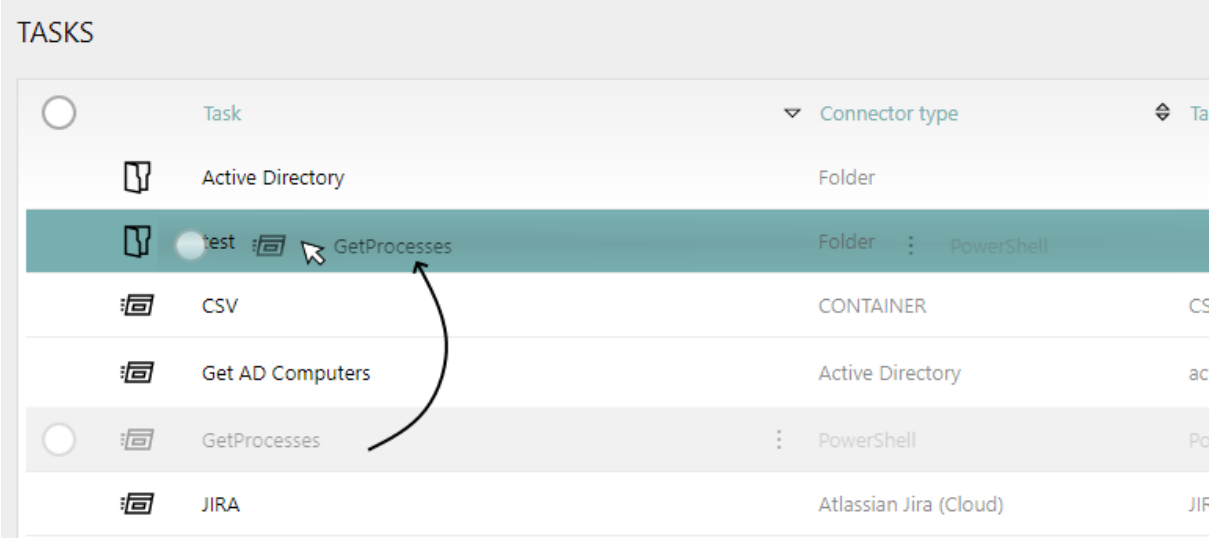
Moving Tasks

Task can be moved by using different techniques:

- Moving Tasks Using Drag-and-Drop
- Moving Tasks Using the Context Menu

Moving Tasks Using Drag-and-Drop

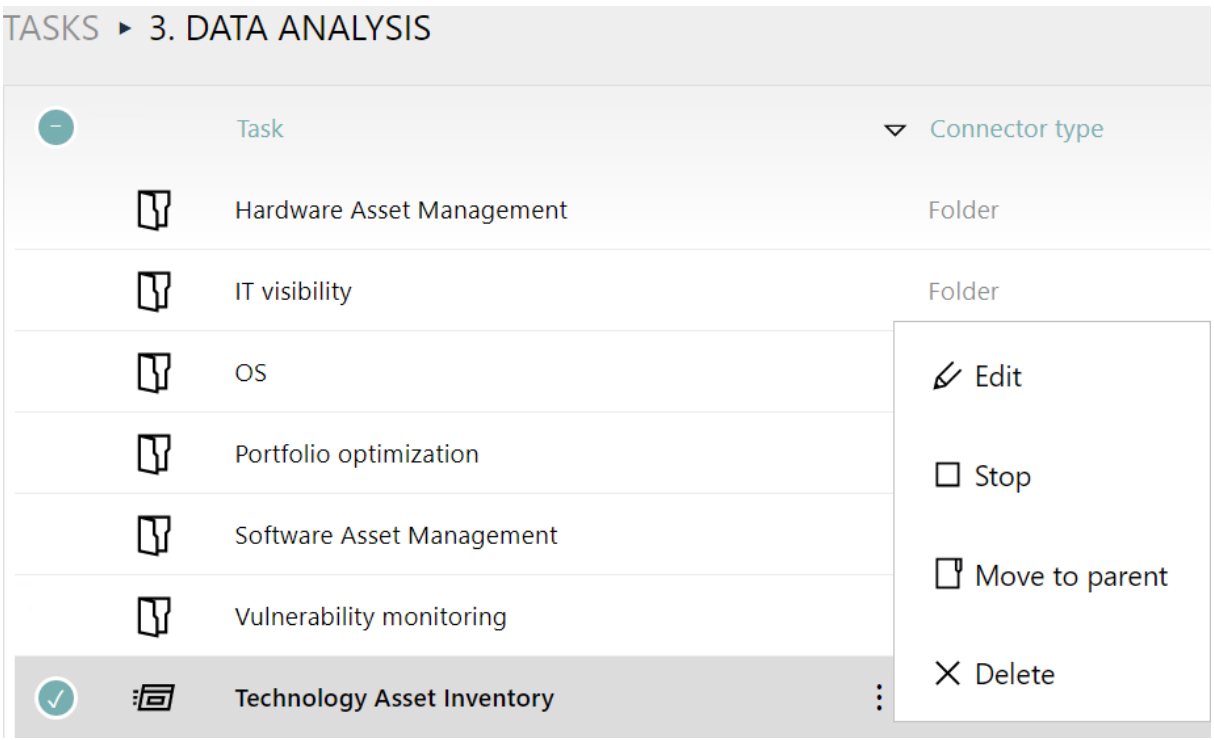
Tasks can be ordered (moved) to a subfolder by using a drag-and-drop technique. The dragging is started after pressing and holding the left mouse button, once the cursor is directly over the task to be dragged (note: the task name is a link, and is not draggable - to drag make sure the cursor is not hovering the link).



Use this technique to move a task into a subfolder. In order to move a task to its parent folder, use the context menu.

Moving Tasks Using the Context Menu

Tasks can be moved to their parent folder by using the context menu. Open the context menu by clicking on the three dots ⋮ that will appear when hovering over the task.



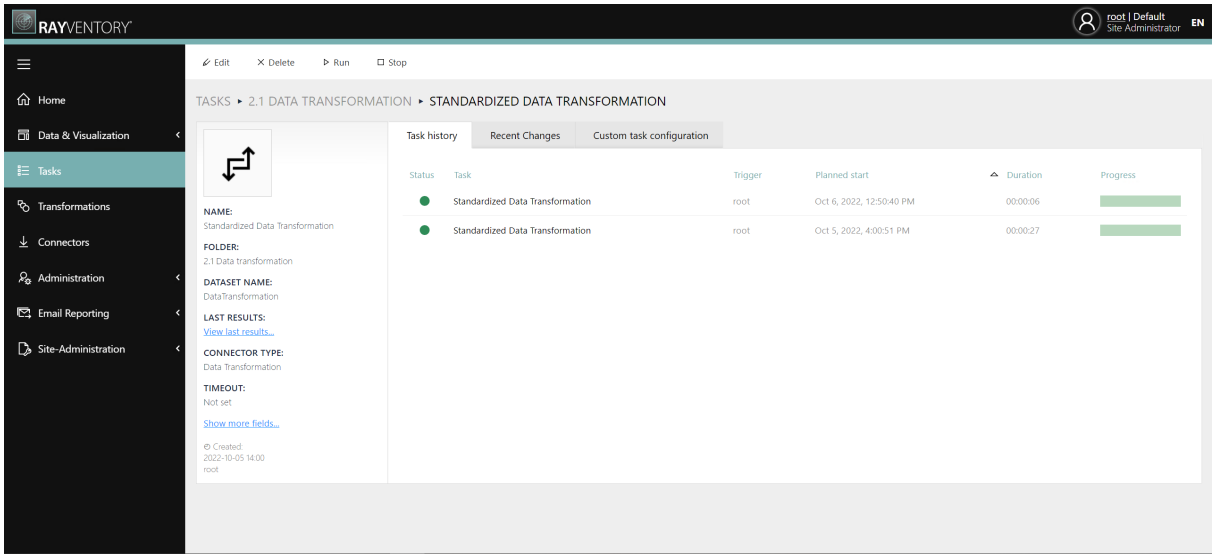
Click on the **Move to parent** option in the context menu, in order to move the task to the parent folder. To move a task into a subfolder use the drag-and-drop technique.

Task Details

Clicking on the name of a task in the tasks view navigates to the task details. This view shows the basic details, configuration and recent history of a task.

The task can be edited and deleted using the action buttons located in the upper horizontal action bar:

- **Edit**
  Opens the "Edit Task" panel that allows it to edit the current task details
- **Delete**
  Deletes the current task
- **Run**
  If the selected task can be started (its type is licensed, there is a valid agent configuration and all settings are in place) this button starts it on demand, ignoring any schedule set on the task.
- **Stop**
  This option is active for tasks that are currently running. Pressing this button and confirming the action stops the task and rejects any results it collected so far.



There are three tabs available in this view:

- **Task History**
  Shows the list of recent runs, timing and their status. Clicking on a name of the task shows extra details and status.

- **Recent changes**
  This view tracks the changes of task configuration.

- **Custom task configuration**
  Shows task configuration parameters, in a JSON format. This view may be replaced with more specialized view for certain connector types (for example MS SQL Server or ODBC).

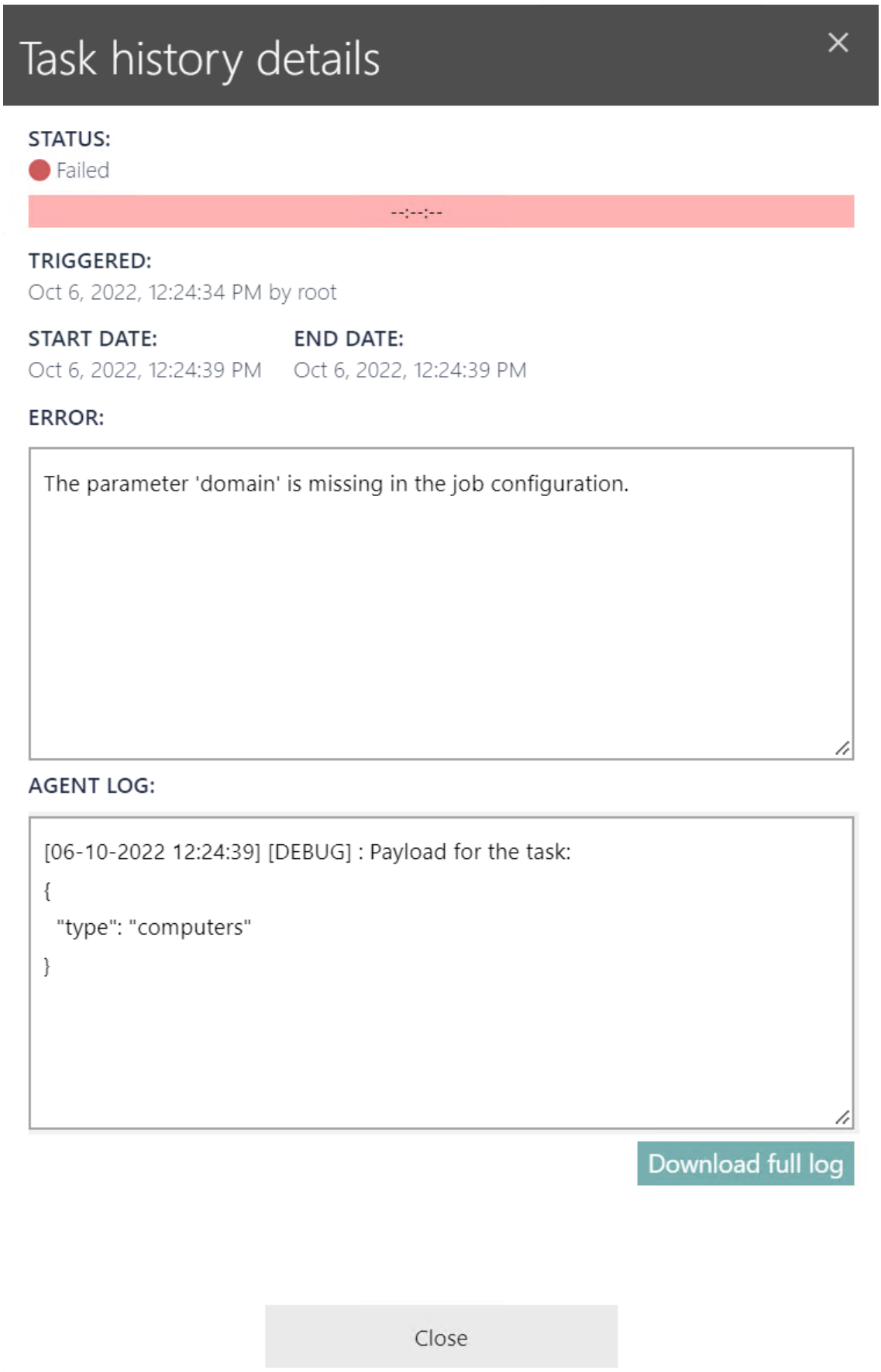
Task History

A task can have multiple statuses as shown in the task history table in the first column. Hovering over the the status shows a tooltip with the current status.

Status	Description
Queued	The task is queued and ready to be fetched by the assigned agent.
Pending	The task was successfully fetched by the assigned agent and is waiting for execution.
Active	The task is currently being executed.
Success	The task was successfully executed and the result table has been written to the Result-database.
Failed	An exception occurred during the execution of the task query. The error message is shown the task history details panel.
Timeout	The execution timeout has expired. The timeout period was either exceeded during the execution of the task or the agent did not respond within the specified task timeout. This status only reached when the timeout was specified on a task.
Expired	The task could not be started within the first 30 minutes of the planned start date.
Canceled	The task was stopped by the user himself.

Task History Details

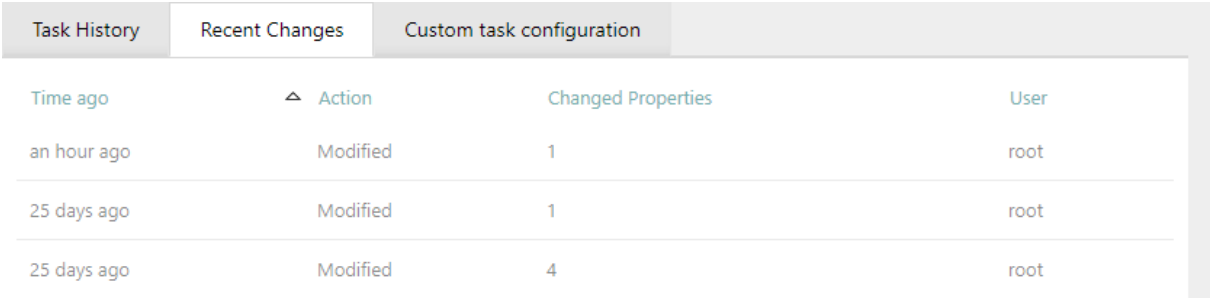
Clicking on a row in the task history table opens its details in a sidebar panel. If, for example, an error occurs, the error message is displayed in this panel, as shown below:.


Task history details
STATUS:
Failed
--:--:--
TRIGGERED:
Oct 6, 2022, 12:24:34 PM by root
START DATE:
Oct 6, 2022, 12:24:39 PM
END DATE:
Oct 6, 2022, 12:24:39 PM
ERROR:
The parameter 'domain' is missing in the job configuration.
AGENT LOG:
[06-10-2022 12:24:39] [DEBUG] : Payload for the task:
{
"type": "computers"
}
Download full log
Close

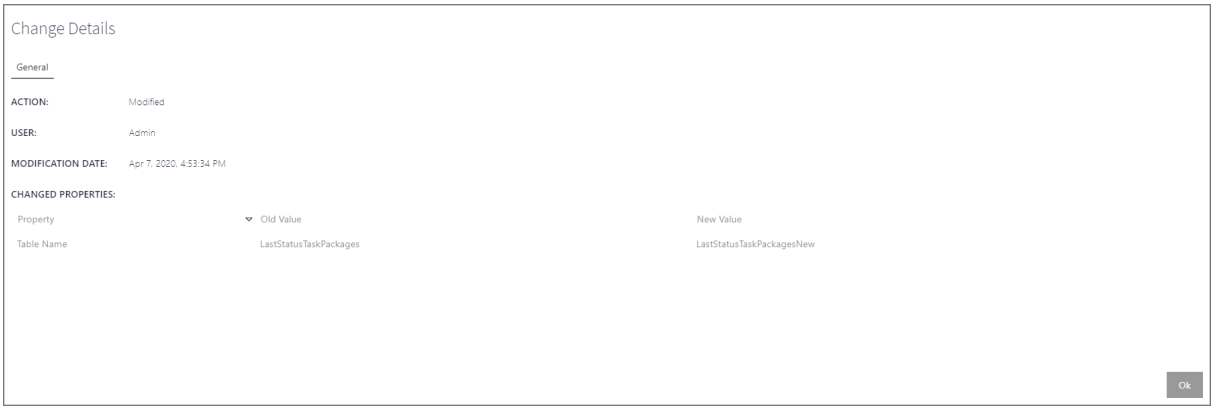
It is possible to download the full logfile for the selected error message by clicking the **Download full log** button. After clicking on the button, the log will automatically be downloaded and can be found in the local **Downloads** folder.

Comparing Task Changes

Select the **Recent Changes** tab from the tab selection box shown in the task details view. A table listing all recent changes made on the task details are shown. To get more details about a recent change click on a table row.



A new modal opens showing detailed information about the values that have been changed by showing the previous and changed value at this time.



If changes were made to the query, these can be viewed in the **Query** tab. The query tab provides a detailed comparison of the changes per line.

Required Permissions

In order to run tasks from RayVentory Data Hub on the RayVentory Server Database it is necessary to grant execution privileges to the following stored procedures:

- csp_aspera_connector_device
- csp_aspera_connector_device_provider_types
- csp_aspera_connector_device_providers
- csp_aspera_connector_device_relation
- csp_aspera_connector_device_relation_types
- csp_aspera_connector_device_types
- csp_aspera_connector_software_arp
- csp_aspera_connector_software_file
- csp_aspera_connector_software_generic_microsoft
- csp_aspera_connector_software_generic_oracle
- csp_aspera_connector_software_generic_os
- csp_aspera_connector_software_generic_othersw
- csp_aspera_connector_software_generic_unixspecial
- csp_aspera_connector_software_msi
- csp_aspera_connector_tag
- CompUsage
- sp_OracleInstances
- sp_rp_OracleVirtualInfrastructure
- f_HardwareProperties

The following `.sql` script can be executed on the RayVentory database in order to grant the necessary execution rights for all tasks:

```
GRANT EXECUTE ON csp_aspera_connector_device TO [AIO\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_device_provider_types TO [AIO
\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_device_providers TO [AIO
\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_device_relation TO [AIO
\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_device_relation_types TO [AIO
\serviceUserDB]
```

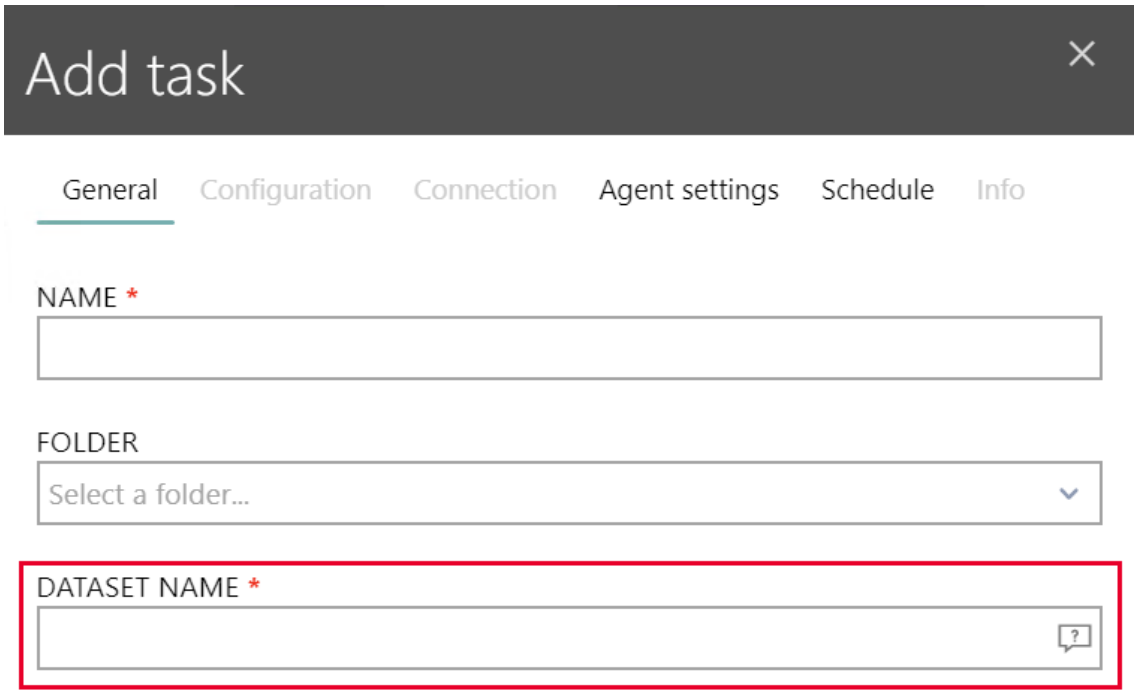
```
GRANT EXECUTE ON csp_aspera_connector_device_types TO [AIO\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_arp TO [AIO\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_file TO [AIO\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_generic_microsoft TO [AIO
\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_generic_oracle TO [AIO
\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_generic_os TO [AIO
\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_generic_othersw TO [AIO
\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_generic_unixspecial TO [AIO
\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_msi TO [AIO\serviceUserDB]
GRANT EXECUTE ON csp_aspera_connector_software_tag TO [AIO\serviceUserDB]
GRANT EXECUTE ON CompUsage TO [AIO\serviceUserDB]
GRANT EXECUTE ON sp_OracleInstances TO [AIO\serviceUserDB]
GRANT EXECUTE ON sp_rp_OracleVirtualInfrastructure TO [AIO\serviceUserDB]

GRANT SELECT ON f HardwareProperties TO [AIO\serviceUserDB]
```

Naming Conventions

The following naming conventions are automatically applied by RayVentory Data Hub when creating tables.

Naming Convention for Tables by an Automated/Transformation Task

For tables created by automated task or transformations, the following naming conventions are applied:



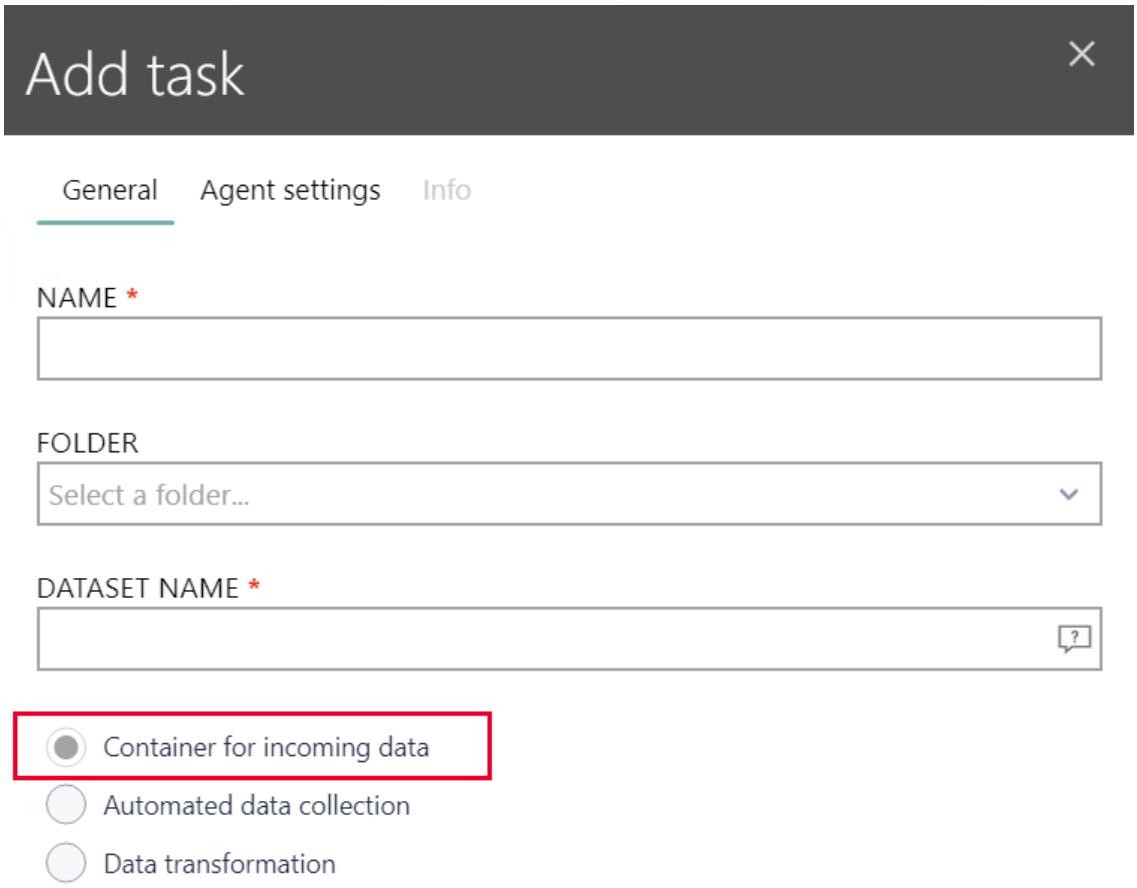
1. If a task only delivers a single CSV to the backend, the backend will create a table with the name that has been entered into the **DATASET NAME** field.
   The name contained in the **DATASET NAME** field may have a maximum of 60 characters and is not allowed to contain the following special characters: . **-**, **"**, **'**, **[**, and **]**.
   - **Example**: `dbo.TaskDataSetname`

2. If a task delivers more than one CSV file to the backend, the backend will create multiple tables with the same prefix and different suffixes. Prefix and suffix are **always** delimited by the **-** (minus) symbol. The prefix is **always** the dataset name of the task followed by a **-** (minus) and the suffix. The suffix is **always** the the filename of the CSV file sent by the agent.
   - **Example**: `dbo.TaskDataSetname-Filename`

Note:
It is possible for a user to change the prefix by changing the dataset name.
It is **not** possible for a user to change the suffix since this is determined by the file name.

Naming Conventions for Tables Created by Importing Files into a Container

When importing CSV, XLS, or XLSX files into a container, the following naming conventions apply.



Importing CSV-Files

When importing a single CSV file, the name of the table will be identical to the dataset name. The dataset name can have a maximum of 60 characters and is not allowed to contain the following special characters: . -, ", ', [, and].

- **Example**: `dbo.TaskDataSetname`

If more than one CSV file is imported, multiple tables with a prefix and a suffix will be created. The prefix is **always** the dataset name of the task followed by a **-** (minus) and the suffix. The suffix is **always** the the filename of the CSV file.

- **Example**: `dbo.TaskDataSetname-Filename`

Note:
It is possible for a user to change the prefix by changing the dataset name.
It is **not** possible for a user to change the suffix since this is determined by the file

name.

Importing XLS- or XLSX-Files

When importing one Excel file the following naming convention is applied.

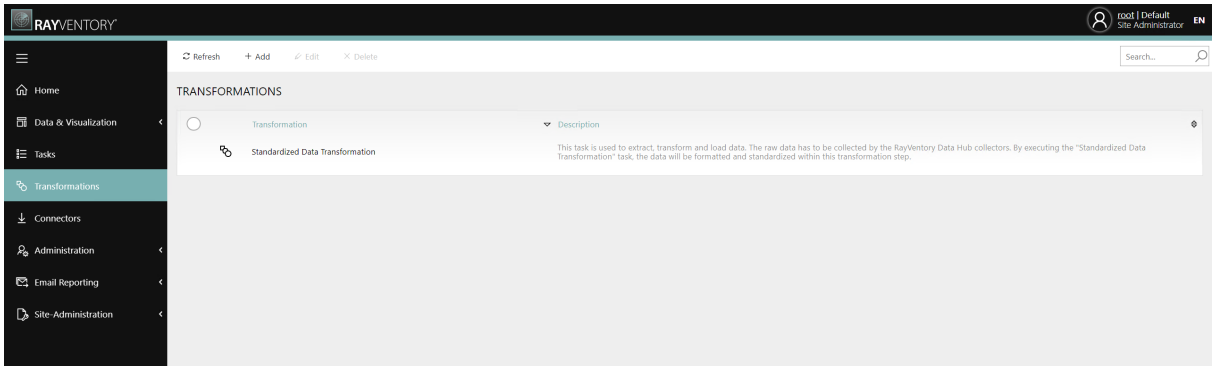
- If the file only contains one sheet, the tablename will be identical to the dataset name of the task.
  - **Example:** `dbo.TaskDataSetname`
- If the file contains multiple sheets, multiple tables will be created. The resulting table names will look as follows: the dataset name of the task followed by a - (minus), then the filename followed by a **-** (minus), and then the name of the sheet.
  - **Example:** `dbo.TaskDataSetname-Filename-Sheetname`

When importing multiple Excel files the following naming convention is applied.

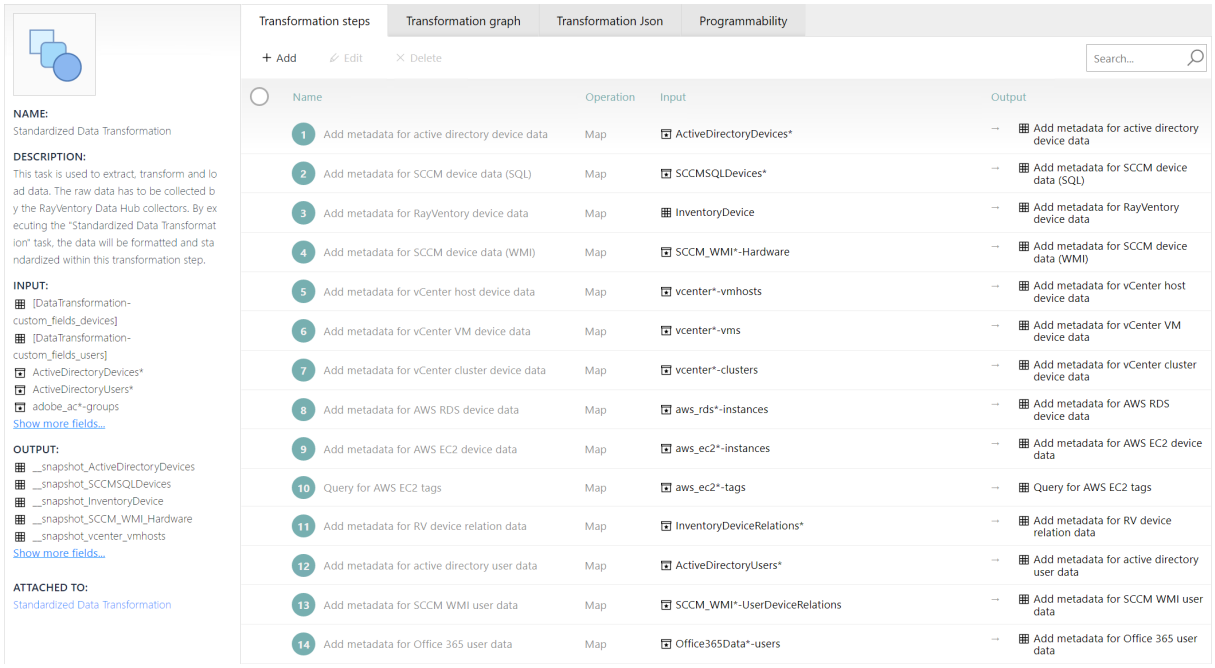
- For files that contain only one sheet, the tablename will be the dataset name of the task followed by a **-** (minus) and the filename.
  - **Example:** `dbo.TaskDataSetname-Filename`
- For files that contain multiple sheets, multiple tables will be created. The resulting table names will look as follows: the dataset name of the task followed by a **-** (minus), then the filename followed by a **-** (minus), and then the name of the sheet.
  - **Example:** `dbo.TaskDataSetname-Filename-Sheetname`

Transformations

With the ETL technology, it is possible to implement the transformation processes in a configurable way.



On the main page of the Transformations section it is possible to create new transformations and to edit or delete the available transformations. In order to edit or delete a transformation, it is necessary to select the transformation in the list shown in the main area. How to add or import a transformation to RayVentory Data Hub is described in the Adding and Importing Transformations chapter. To view the the details of a specific transformation click on the name of the transformation.



On the left side of the window the general information of the transformation are listed. This part of the screen contain the following information:

- **NAME**: This field contains name of the transformation.

- **DESCRIPTION**: This field may contain a short description of the transformation.
- **INPUT**: The name of the table or tables from which the data is taken.
- **OUTPUT**: The name of the table containing the transformed data.
- **ATTACHED TO**: Contains a list of the tasks using this specific transformation.

The right part of the screen contains three tabs.

The first tab is called **Transformation steps** and contains a list of the transformation steps that are part of the transformation. In order to add a new transformation step, click on the **+ Add** entry on top of the list. It is also possible to edit or delete a selected transformation step by using the two corresponding entries. Information on how to create or edit transformation steps can be found in the Creating and Editing Transformation Steps chapter.

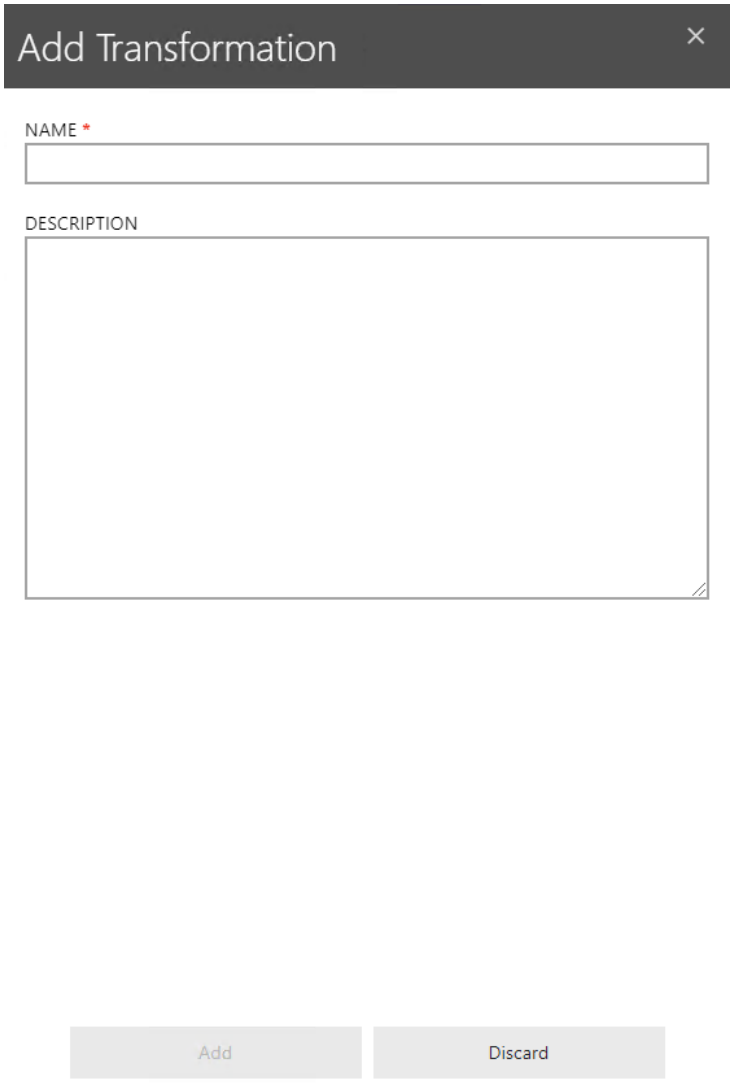
The second tab is called **Transformation Graph** and contains a graph showing a visual representation of the transformation steps. More information can be found in the Transformation Graph chapter.

The third tab is called **Transformation Json** and contains the content of the `.json` file of the transformation steps. More information can be found in the Transformation Json chapter.

The fourth tab is called **Programmability** and can be used to add SQL macros to the transformation. Information on how to use custom SQL macros in transformations can be found in the Adding Custom Marcos to Transformations chapter.

Adding and Importing Transformations

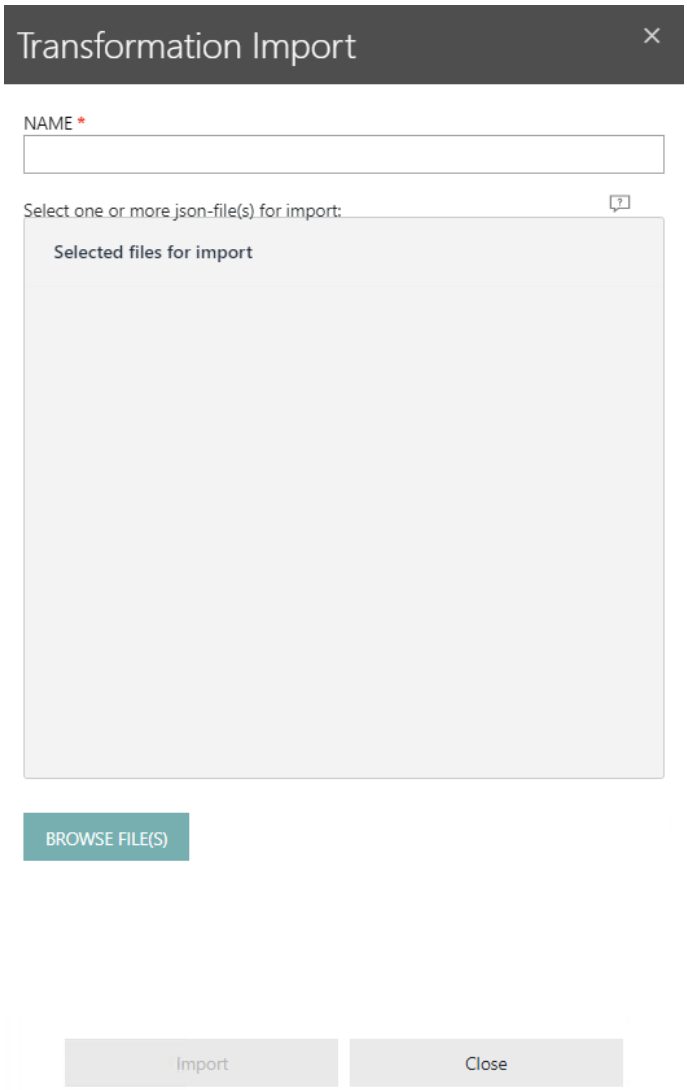
A new tranformation can be added by clicking on the **+ Add** button located at the top of the screen in the **Transformations** screen and selecting the **+ New** option from the dropdown menu. The **Add Transformation** dialog will be opened.



Enter a name for the new transformation into the **NAME** field and enter a short description for the transformation into the **DESCRIPTION** field (the description is optional). Click on the **+ Add** button to create the empty transformation. Now add the necessary transformation steps and custom macros as described in the chapter Creating and Editing Transformation Steps and the chapter Adding Custom Macros to Transformations.

Import a Transformation

A transformation can be imported by clicking the **+ Add** button located at the top of the screen in the **Transformations** screen and selecting the **Import** option from the dropdown menu. It is also possible to import a transformation by clicking the **Import** button at the top of the screen on the details page of a transformation. The **Transformation Import** dialog will be opened.



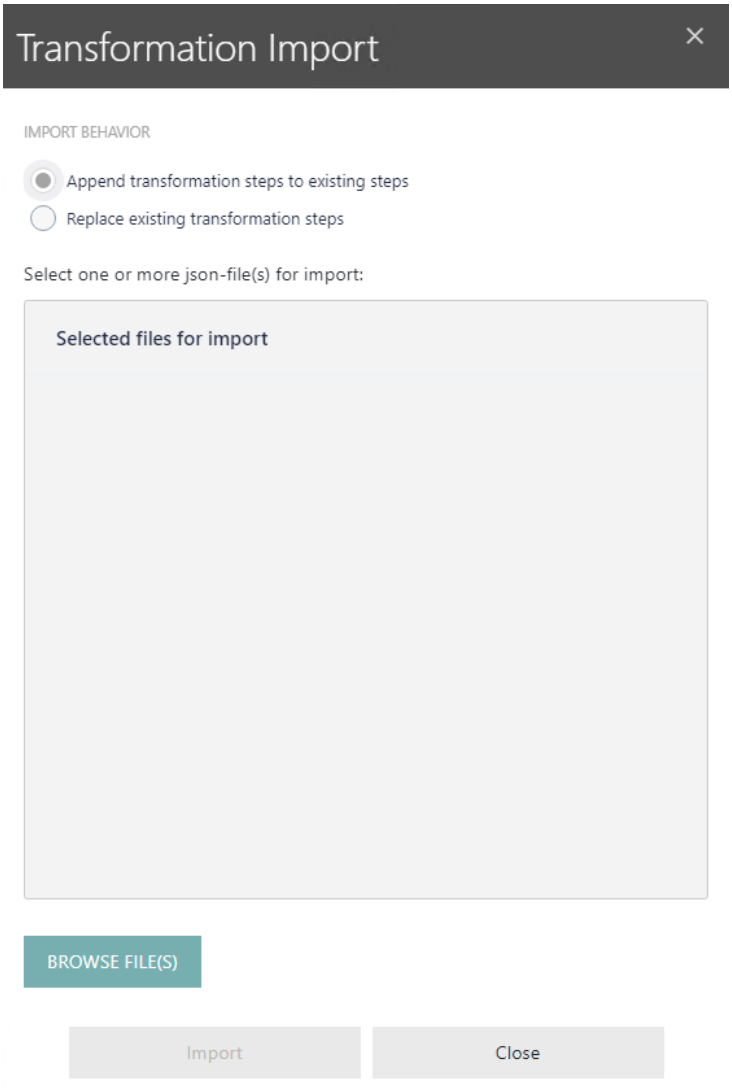
The **Transformation Import** dialog can be used to import transformation files in the JSON format. In order to import a transformation, enter a name for the transformation into the **NAME** field. Click on the **BROWE FILE(S)** button to add one or more `.json` files to the the dialog. When all files have been added, click on the **Import** button to create the transformation and to import the files.

Creating and Editing Transformation Steps

When opening the **Details** screen of a transformation, it is possible to edit the name and the description of the transformation, to delete and export the transformation, and to import transformation step by using the options available in the top bar located above the header. The existing transformation steps will be shown in the list available in the the **Transformation steps** tab. Furthermore it is possible to add, edit, and delete transformations steps by using the buttons located in the **Transformation steps** tab.

Importing Transformation Steps

To import transformation steps to the transformation click the **Import** button at the top of the screen on the details page. The **Transformation Import** dialog will be opened.



The **Transformation Import** dialog can be used to import transformation files in the JSON format. In order to import transformation steps, first select the **IMPORT BEHAVIOR**. There are

two different behaviors available.

- **Append transformation steps to existing steps**: Use this option to add the steps from the selected files to already existing steps.
- **Replace existing transformation steps**: Use this option to replace already existing steps with the steps from the selected files.

Click on the **BROWE FILE(S)** button to add one or more `.json` files to the the dialog. When all files have been added, click on the **Import** button to import the steps from the selected files and add them to the transformation.

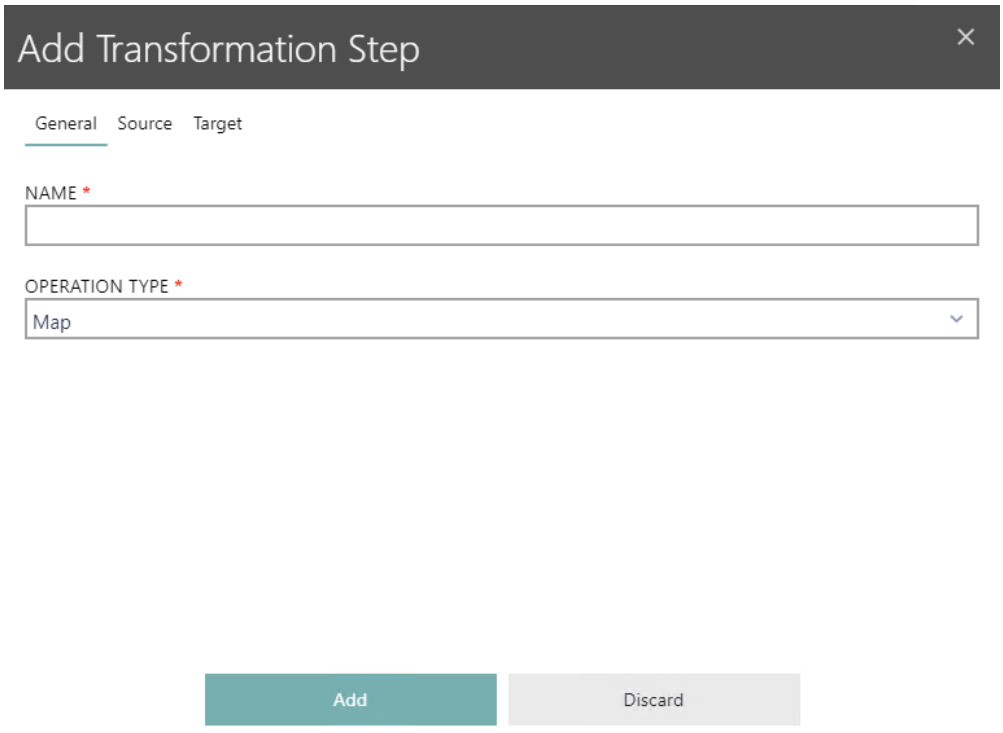
Adding a New Transformation Step and Editing Existing Transformation Steps

To add a transformation step to the transformation, click on the **Add** button in the **Transformation steps** tab. The **Add New Transformation Step** dialog will be shown.

To edit a transformation step, select one of the transformation steps in the **Transformation step** tab and click on the **Edit** button located in the top of the tab. The **Edit Transformation** dialog will be shown.

General

In the **General** tab two fields are available.



- **NAME**: This field contains the name of the transformation step.
- **OPERATION TYPE**: Defines the type of the transformation step that is being configured. The following types are available:
  - Map
  - Join
  - Filter
  - Group
  - Deduplicate
  - Split
  - Enrich

Source

The content of the **Source** tab depends on the type chosen in the **OPERATION TYPE** field of the **General** tab.

Map

Mapping is a process which covers one of the following use cases:

- An output table for a single input table is created. The new table has the exact number of rows as the old one, but the columns may be different.
- One output table is created from several input tables. The new table has the exact number of rows as all of the selected tables together and it also contains all of the unique columns from the selected tables.

The mapping steps requires that the user defines the list of columns to be written in the new table. There are three ways to do this:

- By specifying all required columns.
- By skipping the specification of the required columns but setting the `mapRemaining` attribute to **`true`**.
- By using both together - specifying only the columns which will be transformed and using `mapRemaining` to infer the remaining, undefined columns and include them as well.

A column may be taken-over or transformed. The following mappings are available:

- **Simple mapping**: A column may be simply taken over without any additional processing (simple mapping). The name may be taken as is or changed to an arbitrary game.
- **Fixed values**: A new column may be created containing fixed values.
- **Auto values**: A new column may be created containing values inserted dynamically (random numbers, date and time, GUIDs, placeholders).
- **Transformed values**: A new column may be created by transforming the existing column using a set of transform options (uppercasing, lowercasing, switch-case statements, etc.).
- **Aggregated values**: A new column my be created by aggregating two or more other columns (max/min value, average, concatenated string, first not-null value, etc.).
- **Custom values**: A custom value may be calculated using SQL syntax.

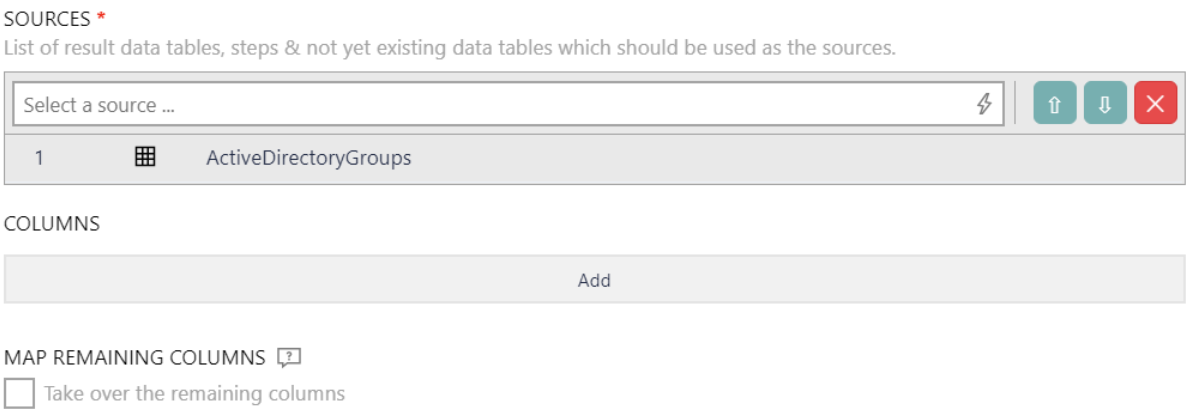
Mapping uses the following syntax:

```
{
   "id": 1, // unique ID
   "type": "map",
   "name":  "Description of the step",
   "source": "Name of the source table",
   "columns": {
      // a dictionary of columns
      "TargetName1": {}, // definition of source1
      "TargetName2": {} // definition of source2
      [...]
   },
   "target": "The name of the output table"
}
```

The following minimum properties are required:

- ID (must be unique)
- Type (must be set to "map")
- Source (must be one of the following)
  - A string representing the table name.
  - A string representing a wildcard to look for table(s).
  - An integer representing the source as another step.
  - An object with the property `table` set to the name of the source table.
  - An object with the property step set to the ID of the source step.
  - An array of tables or steps to perform union select.
- Either a non-empty list of columns or the attribute `mapRemaining` set to `true`.

Configure Mapping in RayVentory Data Hub

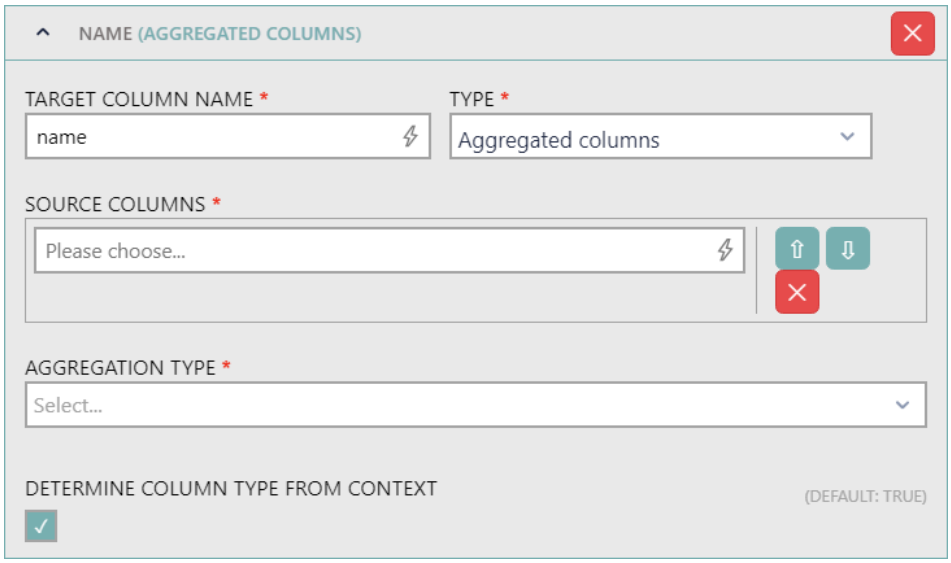


- **SOURCES**: Define one or more sources that will be used for the transformation. It supports auto-completion and will offer all available tables matching the current input string for selection.
- **COLUMNS**: Enter a column from which the value will be taken by clicking on the Add button

and choosing a column and a type as described below for the different types.

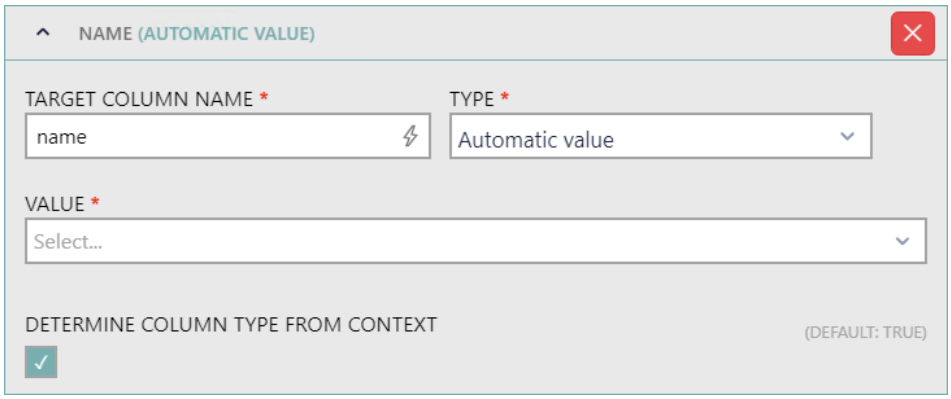
- **MAP REMAINING COLUMNS**: When the field is checked, the remaining columns will be taken over. Disable the option by unchecking the checkbox.

Aggregated Columns



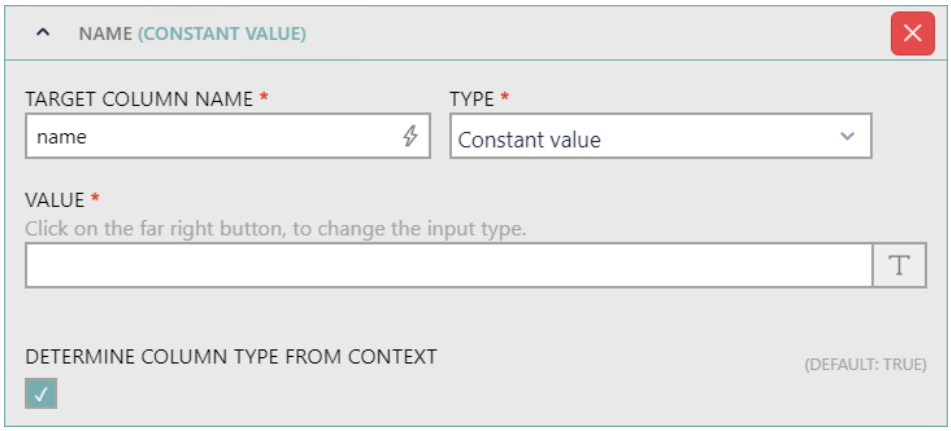
- **TARGET COLUMN NAME**: Enter the name of the target column. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TYPE**: Select **Aggregated columns** to define an aggregation type that will be used with the selected column. If **Aggregated columns** is selected as type, the following fields will be available.
- **SOURCE COLUMN**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **AGGREGATION TYPE**: Define the method that will be used to produce a single value out of the input values by selecting the aggregation type from the dropdown menu. The following options are available.
  - **Average**: Selects the average value of one or more values.
  - **Coalesce**: Selects the first not-empty value from the list of one or more values.
  - **Concat**: Join all given non-empty values using a specific separator (from left to right).
  - **First non null**: Selects the first not-empty value from the list of one or more values.
  - **Maximum**: Selects the maximum of one or more values.
  - **Minimum**: Selects the minimum of one or more values.
  - **Sum**: Selects the sum of one or more values.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

Automatic Value



- **TARGET COLUMN NAME**: Enter the name of the target column. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TYPE**: Select **Automatic value** to define an automatic value that will be used. If **Automatic value** is selected as type, the following fields will be available.
- **VALUE**: Use the dropdown menu to select the automatic value that should be used. The following values are available:
  - **Current datetime**
  - **GUID**
  - **Auto incremented number**
  - **Table name**
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

Constant Value

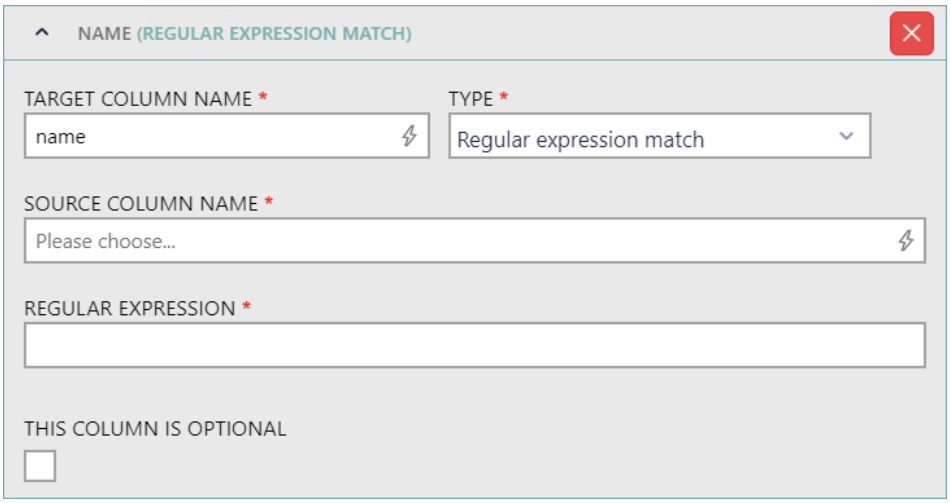


- **TARGET COLUMN NAME**: Enter the name of the target column. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TYPE**: Select **Constant value** to define a value that will be used. If **Constant value** is selected

as type, the following fields will be available.

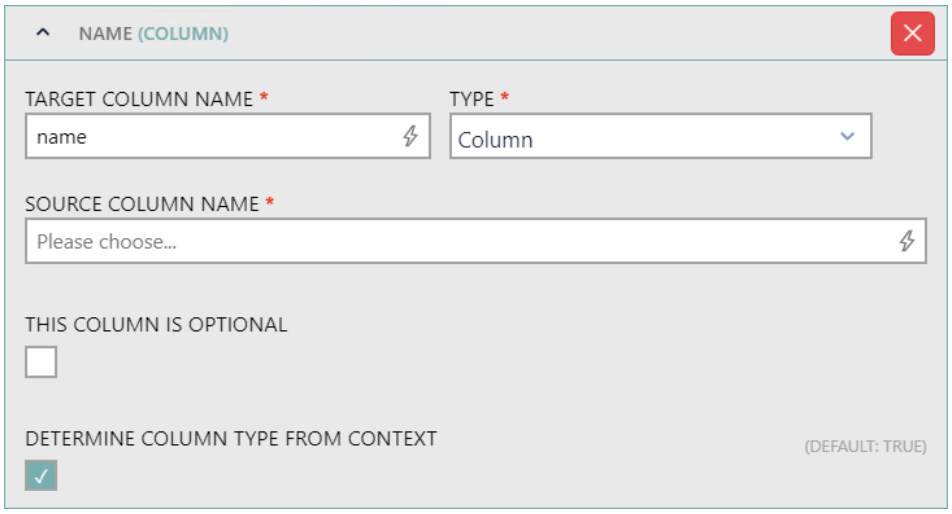
- **VALUE**: Enter the value that should be used. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

Regular Expression Match



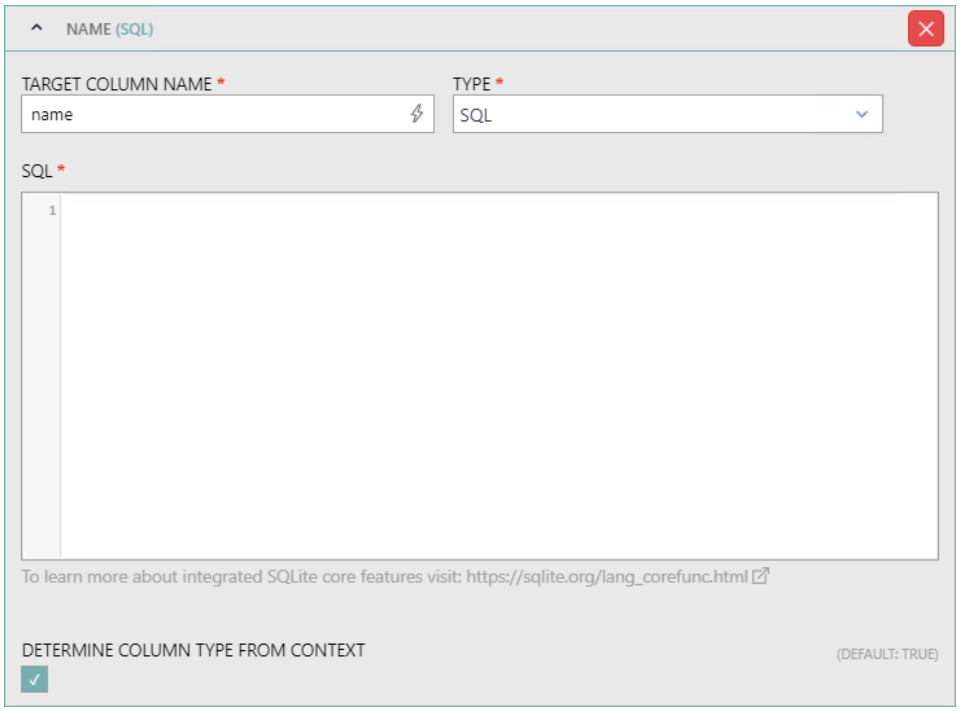
- **TARGET COLUMN NAME**: Enter the name of the target column. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TYPE**: Select **Regular expression match** to define a regular expression that will be used to transform the column. If **Regular expression match** is selected as type, the following fields will be available.
- **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **REGULAR EXPRESSION**: Enter the regular expression that will be used for the transform. Detailed information on how to use regular expressions in ETL can be found in the *Programmability* chapter in the *ETL Implementation Guide*.
- **THIS COLUMN IS OPTIONAL**: If this option is not checked, the ETL engine will check if the source column exists before starting the execution. If the option is checked, the engine will start the execution even if the column does not exist.

Column


NAME (COLUMN)
TARGET COLUMN NAME *
name
TYPE *
Column
SOURCE COLUMN NAME *
Please choose...
THIS COLUMN IS OPTIONAL
DETERMINE COLUMN TYPE FROM CONTEXT
(DEFAULT: TRUE)

- **TARGET COLUMN NAME**: Enter the name of the target column. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TYPE**: Select **Column** to define a column that will be used. If **Column** is selected as type, the following fields will be available.
- **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **THIS COLUMN IS OPTIONAL**: If this option is not checked, the ETL engine will check if the source column exists before starting the execution. If the option is checked, the engine will start the execution even if the column does not exist.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

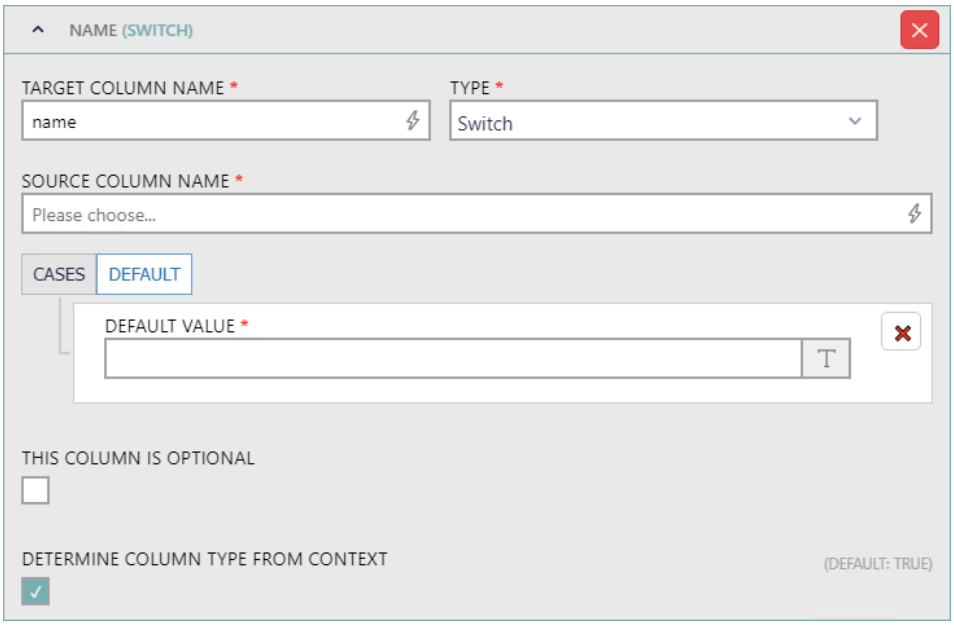
SQL



- **TARGET COLUMN NAME**: Enter the name of the target column. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TYPE**: Select **SQL** to define a macro that will be used to transform the column. If **SQL** is selected as type, the following fields will be available.
- **SQL**: Enter the SQL macro that will be used for the transform.
  Detailed information on how to use scripts in ETL can be found in the *Programmability* chapter in the *ETL Implementation Guide*.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the

COLUMN TYPE field.

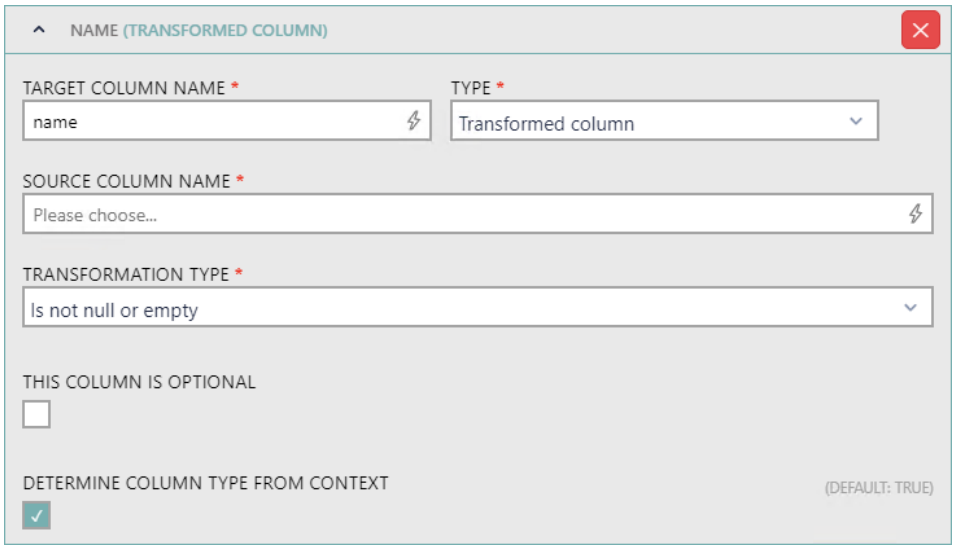
Switch



- **TARGET COLUMN NAME**: Enter the name of the target column. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TYPE**: Select **Switch** to define a switch that will be used to transform the column. If **Switch** is selected as type, the following fields will be available.
- **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **CASES**: If the **CASES** tab is selected, instead of the **DEFAULT** tab, it is possible to add one or more cases. If a case is added the following fields will be available for configuration:
  - **CASE**: Enter a value that should be used for the case. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
  - **THEN**: Enter a value that should be used as then. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
- **DEFAULT**: If the **DEFAULT** tab is selected, instead of the **CASES** tab, it is possible to add one default value by choosing default. Then a new field will be available where the value can be defined.
  - **DEFAULT VALUE**: Enter a value that should be used default. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
- **THIS COLUMN IS OPTIONAL**: If this option is not checked, the ETL engine will check if the source column exists before starting the execution. If the option is checked, the engine will start the execution even if the column does not exist.

- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

Transformed Column



- **TARGET COLUMN NAME**: Enter the name of the target column. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TYPE**: Select **Transformed column** to define a transformation type that will be used to transform the column. If **Transformed column** is selected as type, the following fields will be available.
- **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TRANSFORMATION TYPE**: Select the transformation type that will be used to transform the column.
  - **Is not null or empty**
  - **Is null or empty**
  - **Length**
  - **Lowercase**
  - **Uppercase**
- **THIS COLUMN IS OPTIONAL**: If this option is not checked, the ETL engine will check if the source column exists before starting the execution. If the option is checked, the engine will start the execution even if the column does not exist.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

Join

Joining is an operation which takes two or more tables and combines them to a single one. The combination of rows is performed using a special column or set of columns called joining keys. They should be locally unique.

Since two tables may but do not have to have the same rows and keys, conflicts may arise:

- The left table may have some unmatched rows in the right table.
- The right table may have some unmatched rows in the left table.

There are three basic strategies that can be chosen:

- **Outer joining**
  This will combine all matching rows and it will take all unmatched rows as they are, by default using NULL values where values are unmatched.
  For example:
  A list of devices is scanned by system A and system B may have a lot of entries in common. Some devices may be present only in system A and some devices are only in system B. Outer join will ensure that all devices are present in the target table, even the unmatched devices from system A and the unmatched devices from system B.
- **Left joining**
  This will treat the left column as a master table and write all rows which were matched between left and right and any unmatched rows from the left table. Unmatched rows from the right column will be discarded.
  For example:
  The left table contains a list of customers and the right table contains a list of addresses, than the result table will include all the customers but only the address entries for the customers that are listed in the left table. Unmatched addresses will be discarded.
- **Inner joining**
  This will write only rows which exist in both tables. Unmatched rows from the left table and the right table will be ignored.
  For example:
  The left table contains names and the right table surnames. The output table only has names and surnames. All entries where there is only a name or only a surname will be discarded.

The Join step requires a bit of extra information about the key used to join (ID). This key must be present in all joined tables otherwise an exception will be thrown.

It is also possible to join more tables at once (within the same strategy).

Joined tables are by default optional, meaning that the tables do not have to be existing. If any table is missing it will not be joined. If only a single table exists, then it is taken as is and returned as the output. If all the tables are missing, then the step will not be executed and its target table will not be written. There is a way to define required tables, in which case in a missing source table the step will not be executed and its target table will not be written. There is a way to define required tables in which case in case of a missing source table the step will fail and report an error.

Cell Merging

The join operation uses a convention based approach. When defining a join, the information required is:

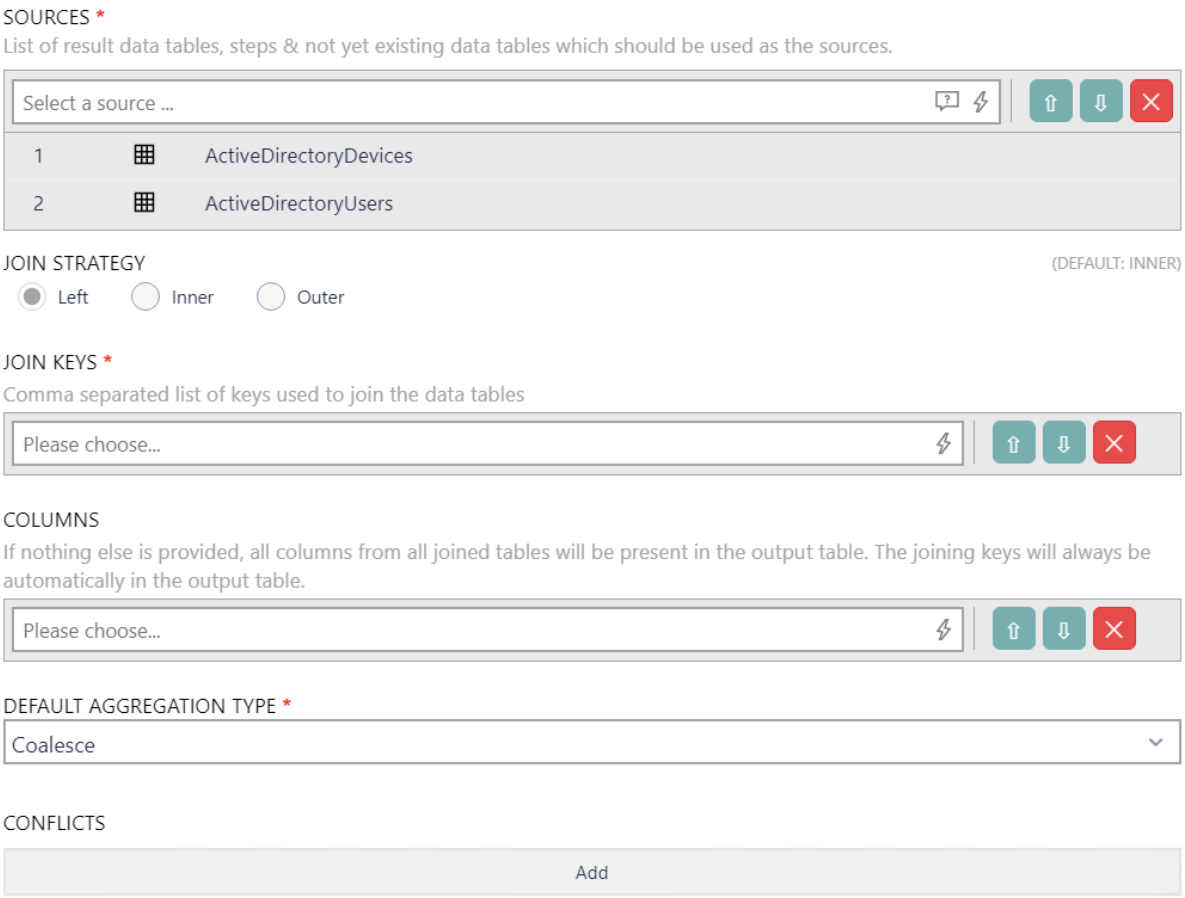
- Which tables need to be joined.
- Which columns are the key used to join them.

All tables taking a part in the join must have all the columns specified as the keys. For other columns the following logic is applied:

- If a column exists in exactly one of the joined tables, it is taken as is.
- If a column exists in two or more of the tables, the values in it are merged using a specific aggregation strategy. The default aggregation is to join the values with a pipe "|" separator.

More information about Joining can be found in the *ETL Implementation Guide*.

Configure Join in RayVentory Data Hub



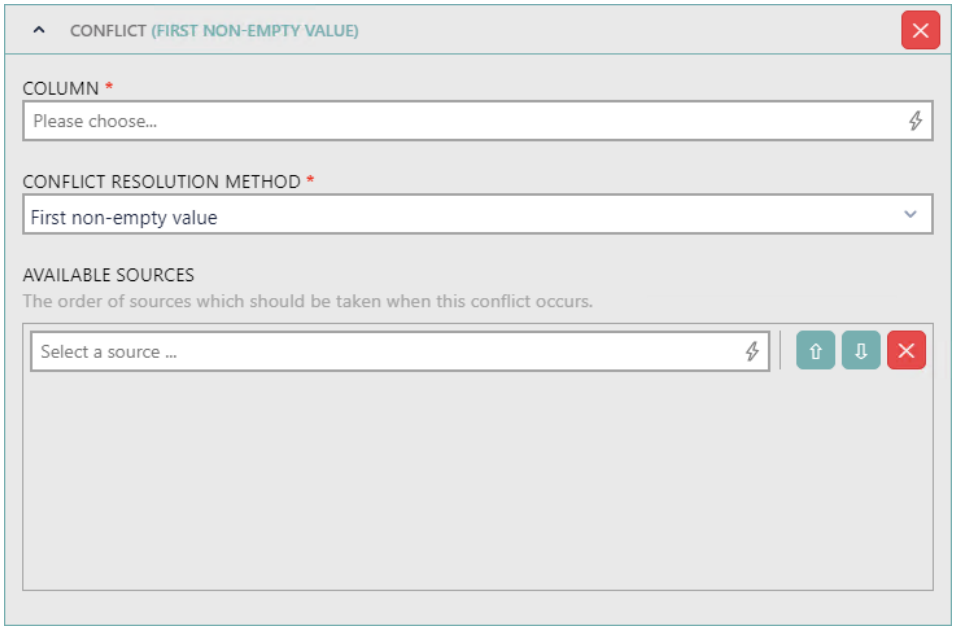
- **SOURCES**: Define one or more sources that will be used for the transformation. It supports auto-completion and will offer all available tables matching the current input string for selection.
- **JOIN STRATEGY**: Select the join strategy that will be used for the transformation. The following strategies are available:
  - **Left**: The left table will be used as master table. All matched rows will be combined and all unmatched rows from the left table will be written. All unmatched rows from the right table will be ignored.
  - **Inner**: All matched rows will be combined and all unmatched rows will be ignored.
  - **Outer**: All matching rows will be combined and all unmatched rows will be taken as they

are.

- **JOIN KEYS**: Enter the join keys used for the join. The key used to join must be present in all joined tables.
- **COLUMNS**: This field is used to specify the columns that will be present in the output table. If no columns are provided in this field, all columns will be available. The join keys will always be available in the output table.
- **CONFLICTS**: This option is used to define which column and which resolution method is used to resolve conflicts. It is possible to create multiple solution to conflicts. Enter the column into the **COLUMN** field available in the option and select one of the conflict resolution methods from the **CONFLICT RESOLUTION METHOD** field. The available methods are listed below in the *Conflict Resolution Method* section.
- **DEFAULT AGGREGATION**: Define the default aggregation that will be used by selecting a method from the dropdown menu. The following options are available:
  - **Average**: Selects the average value of one or more values.
  - **Maximum**: Selects the maximum of one or more values.
  - **Minimum**: Selects the minimum of one or more values.
  - **Sum**: Selects the sum of one or more values.
  - **Coalesce**: Selects the first not-empty value from the list of one or more values.
  - **Concat**: Join all given non-empty values using a specific separator (from left to right).
  - **First non null**: Selects the first not-empty value from the list of one or more values.

Conflict Resolution Method

First Non-Empty Value

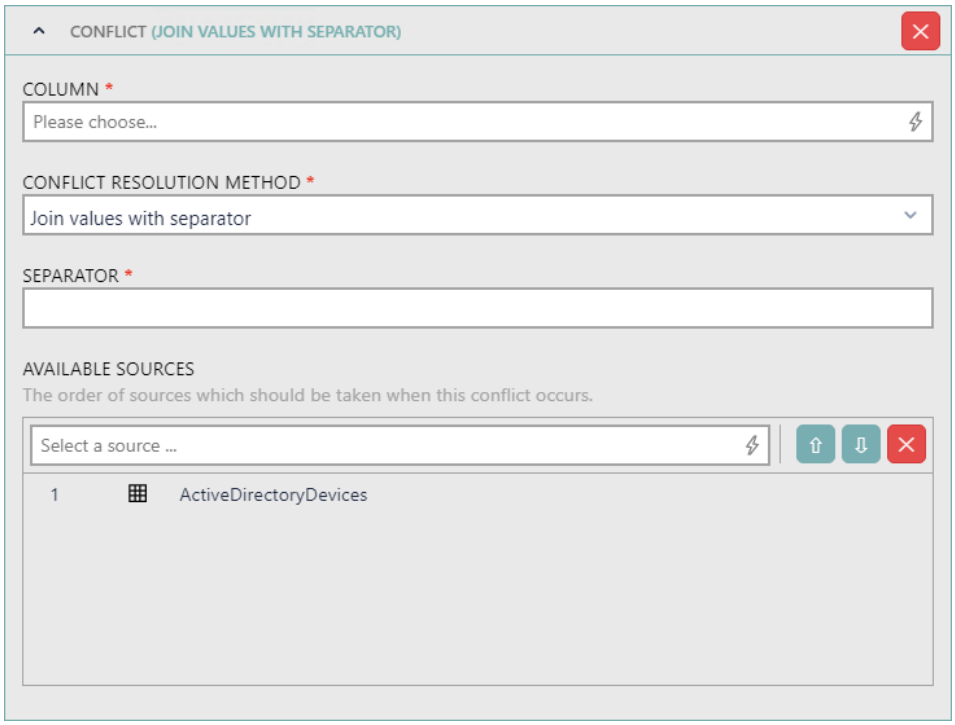


- **COLUMN**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **CONFLICT RESOLUTION METHOD**: Select the method used to resolve conflicts. The available

methods are **First non-empty value**, **Join values with separator**, **Max value**, and **Min value**. If **First non-empty value** is selected, the following further settings will be available.

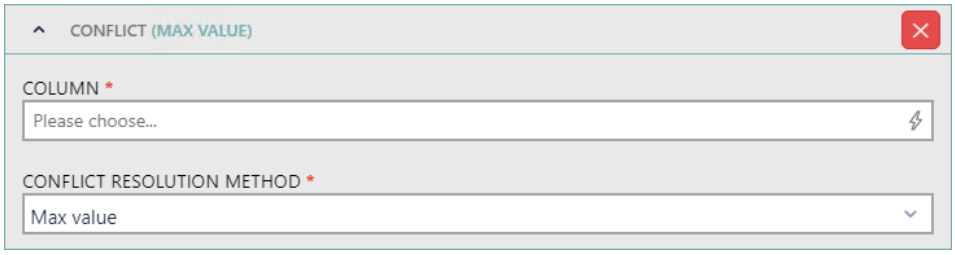
- **AVAILABLE SOURCES**: Ordering the available sources in this field determines the order in which the sources will be taken when this conflict occurs. It supports auto-completion and will offer all available sources matching the current input string for selection.

Join Values with Separator



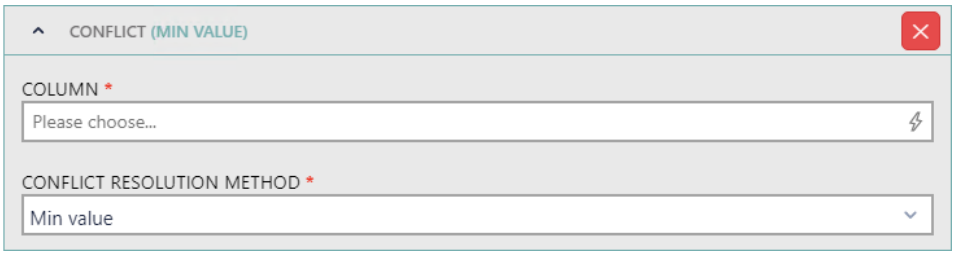
- **COLUMN**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **CONFLICT RESOLUTION METHOD**: Select the method used to resolve conflicts. The available methods are **First non-empty value**, **Join values with separator**, **Max value**, and **Min value**. If **Join values with separator** is selected, the following further settings will be available.
- **SEPARATOR**: Enter the separator that will be used into this field.
- **AVAILABLE SOURCES**: Ordering the available sources in this field determines the order in which the sources will be taken when this conflict occurs. It supports auto-completion and will offer all available sources matching the current input string for selection.

Max Value



- **COLUMN**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **CONFLICT RESOLUTION METHOD**: Select the method used to resolve conflicts. The available methods are **First non-empty value**, **Join values with separator**, **Max value**, and **Min value**. If **Max value** is selected, the following further settings will be available.

Min Value



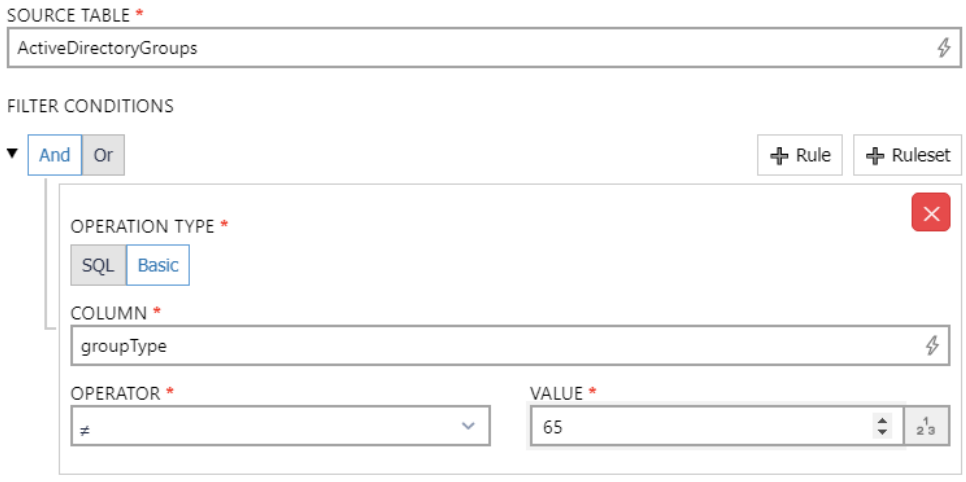
- **COLUMN**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **CONFLICT RESOLUTION METHOD**: Select the method used to resolve conflicts. The available methods are **First non-empty value**, **Join values with separator**, **Max value**, and **Min value**. If **Min value** is selected, the following further settings will be available.

Filter

Filtering takes one value as the input and produces another table with the same schema but with the same or a lower number of rows, based on filter conditions (substraction).

By default, the source is optional, meaning that the table does not need to exist. If the source table is missing, the step will not be executed and its target table will not be written. There is a way to define that the source is required. If this is used, in case of a missing table the step will fail and report an error.

Configure Filter in RayVentory Data Hub



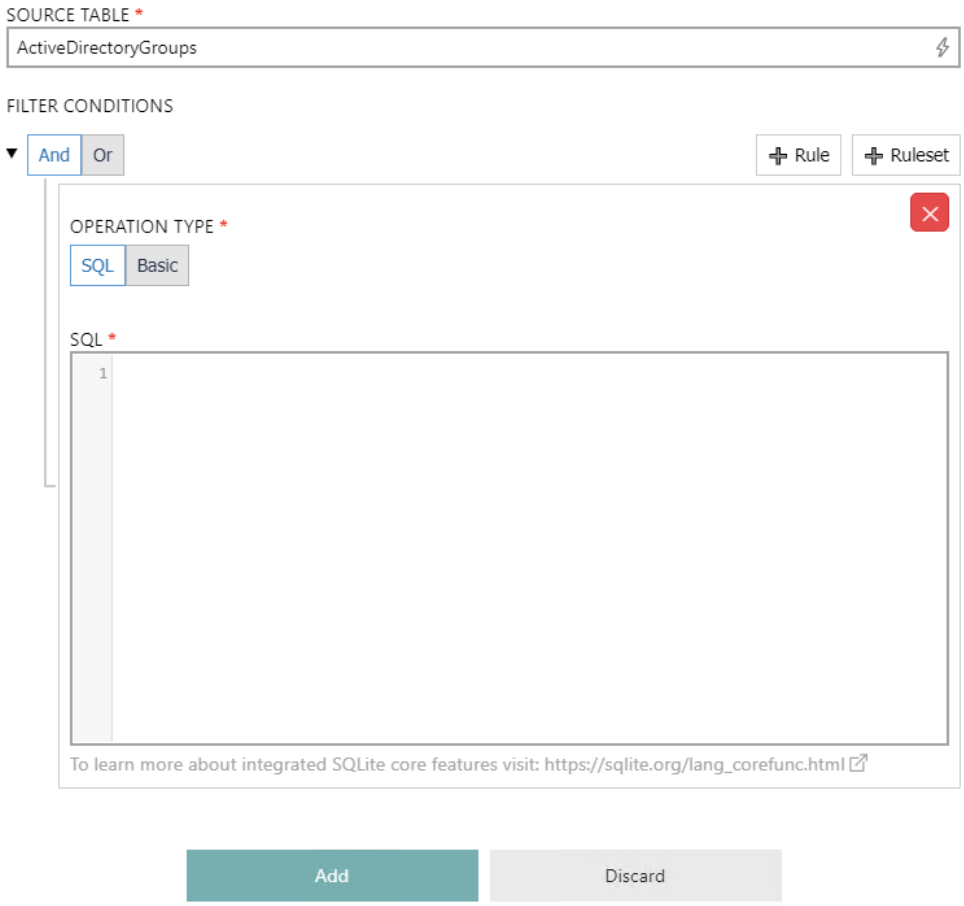
- **SOURCE TABLE**: Enter the table that is used as source for the data. It supports auto-completion and will offer all available tables matching the current input string.
- **FILTER CONDITIONS**: It is possible to add **Filter Conditions** rules and rulesets either as **And** or as **Or** conditions. Select **And** for filter where all conditions of the filter must be met and **Or** for filter where one of the conditions must be met.

- **OPERATION TYPE**: Choose either SQL or Basic as operation type for the rule or ruleset. If SQL is chosen as type, it is possible to use custom SQL macros for the filters. Be aware that if using SQL for filtering, the expression for evaluation should return either `True` or `False`. If Basic has been chosen, there are a couple of fields available for configuration of the filter.

Basic

- **COLUMN**: Enter the column that will be used for the filter. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **OPERATOR**: Select the operator to use for the filter. The following operators are available: **>**, **≥**, **<**, **≤**, **=**, **≠**, **Contains**, **Does not contain**, **Starts with**, **Does not start with**, **Ends with**, and **Does not end with**.
- **VALUE**: Enter the value that should be used. By clicking on the button located on the left side of the field it is possible to switch the input type between text input, number input, and checkbox input.

SQL



- **SQL**: Enter an SQL macro containing the filter into this field. If using SQL for filtering, the expression for evaluation should return either `True` or `False`.
  Detailed information on how to use scripts in ETL can be found in the *Programmability* chapter in the *ETL Implementation Guide*.

Group

Grouping is process which covers the following use cases:

- When having a single table, it writes an output table that has the exact amount of rows but with extra meta-columns containing a "group identifier". Two or more elements considered to be the same have always the same value in the group column. The second extra column being added is the count of rows in the given group.
- When having a single table, it writes an output table that contains aggregated (deduplicated / distinct) rows, where the values in each non-grouped colum is aggregated with a specific function (maximum, minimum, average, concatenated value, coalesced value). An extra column with a number of grouped rows is added.

The following snippet shows a functional example required to group rows and enter the information about the group:

```
{
   "id": 1,
   "type": "group",
   "name": "Example of grouping (deduplicate)",
   "source": "Users",
   "by": [ "Name", "E-Mail" ],
   "target": "Users_Grouped_Deduplicated"
}
```

It is possible to opt-in for deduplication by using the "action" object:

```
{
   "id": 2,
   "type": "group",
   "name": "Example of grouping (deduplicate)",
   "source": "Users",
   "by": [ "Name", "E-Mail" ],
   "target": "Users_Grouped_Deduplicated",
   "action": {
      "type": "deduplicate"
   }
}
```

Required properties:

- ID (must be unique)
- Type (must be set to "group")
- Source (must be one of the following):
  - A string representing the table name
  - An integer representing the source as another step
  - An object with the property "table" set to the name of the source table

- An object with the property "step" set to the ID of the source step

Source is by default optional which means that the table does not have to be existing. IF the table is missing, the step will not be executed and its target table will not be written. There is a way to define that the source is required, in which case in case of a missing source table, the step will fail and report an error. More information about the setup of the required steps can be found in the *ETL Implementation Guide*.

Optional parameters:

- **by**: This defines a single column or a list of columns to be used as grouping keys. Equivalent to the SQL statement `SELECT * FROM <table> GROUP BY <columns>`.
  If this parameter is omitted or set to an empty list, then all columns are used for grouping.
- **target**: If omitted, the output table is temporary and does not get saved during the LOAD phase.
- **action**: The action to execute. This can be one of the following:
  - If the value is a string, then the action of a given type is used. Supported values are deduplicate and recognize. The values are case-insensitive.
  - If the value is an object, then its property type is used to determine the action. The value of the parameter type should be either deduplicate or recognize. The values are case-insensitive. There are some extra properties available when using the object syntax.
  - If the action is omitted, the recognize action is used as the default.

Configure Group in RayVentory Data Hub

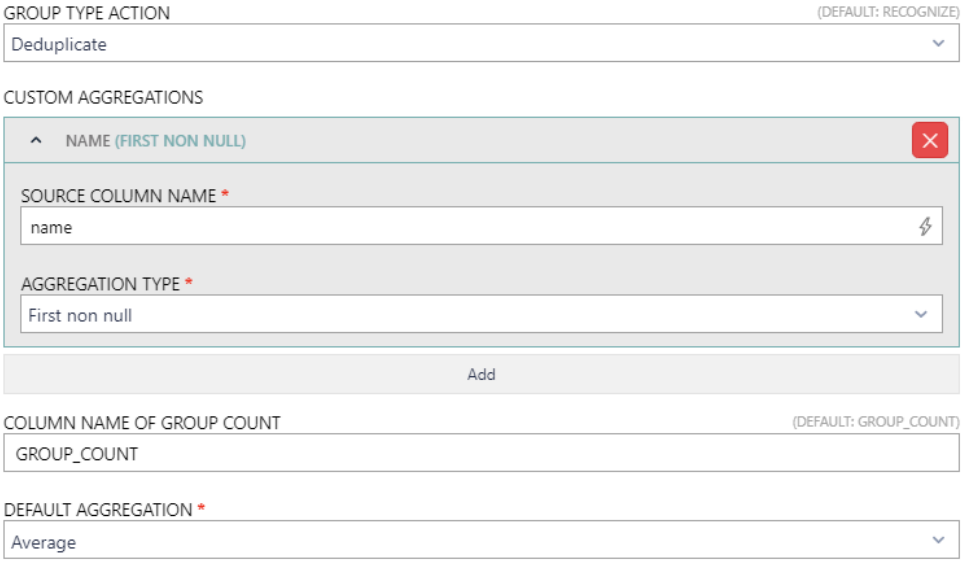
Recognize

- **SOURCE TABLE**: Enter the table that is used as source for the data. It supports auto-completion and will offer all available tables matching the current input string.
- **GROUP BY**: Enter the column names that will be used to group the data. The field supports auto completion and will offer all available columns matching the current input string. The columns will be separated by commas.
- **GROUP TYPE ACTION**: This field is used to select the **Group Type Action** that will be used.

The available options are **Recognize** and **Deduplicate**. The following entries represents the fields that are available if **Recognize** is chosen. For the fields available if **Deduplicate** is chosen see further below.

- **COLUMN NAME OF GROUP KEY**: This field is used to customize the column name of the group key. By default, the name used for the column is GROUP_KEY.
- **COLUMN NAME OF GROUP COUNT**: This field is used to customize the column name of the group count. By default, the name used for the column is GROUP_COUNT.

Deduplicate



- **GROUP TYPE ACTION**: This field is used to select the **Group Type Action** that will be used. The available options are **Recognize** and **Deduplicate**. The following entries represents the fields that are available if **Deduplicate** is chosen. For the fields available if **Recognize** is chosen refer to the previous section.
- **CUSTOM AGGREGATIONS**: This can be used to add additional customized aggregations to the transformation.
  - **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
  - **AGGREGATION TYPE**: Define the method that will be used to produce a single value out of the input values by selecting the aggregation type from the dropdown menu. The following options are available.
    - **Average**: Selects the average value of one or more values.
    - **Coalesce**: Selects the first not-empty value from the list of one or more values.
    - **Concat**: Join all given non-empty values using a specific separator (from left to right).
    - **First non null**: Selects the first not-empty value from the list of one or more values.
    - **Maximum**: Selects the maximum of one or more values.
    - **Minimum**: Selects the minimum of one or more values.
    - **Sum**: Selects the sum of one or more values.
- **COLUMN NAME OF GROUP COUNT**: This field is used to customize the column name of the group count. By default, the name used for the column is GROUP_COUNT.
- **DEFAULT AGGREGATION**: Define the method that will be used to produce a single value out of the input values by default by selecting the aggregation type from the dropdown menu. The following options are available.
  - **Average**: Selects the average value of one or more values.
  - **Maximum**: Selects the maximum of one or more values.
  - **Minimum**: Selects the minimum of one or more values.
  - **Sum**: Selects the sum of one or more values.
  - **Coalesce**: Selects the first not-empty value from the list of one or more values.

- **Concat**: Join all given non-empty values using a specific separator (from left to right).
- **First non null**: Selects the first not-empty value from the list of one or more values.

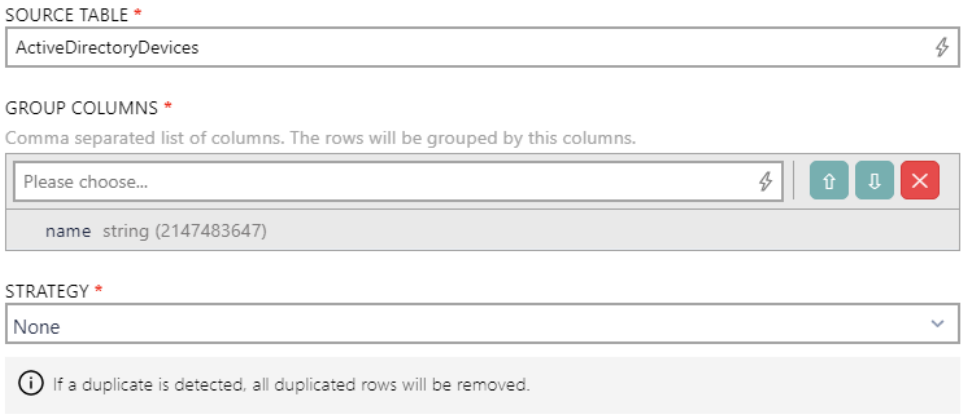
Deduplicate

Deduplicating is a process of taking a table as an input, group the similar records by one or more columns, and then decide on each set how to proceed with the records.

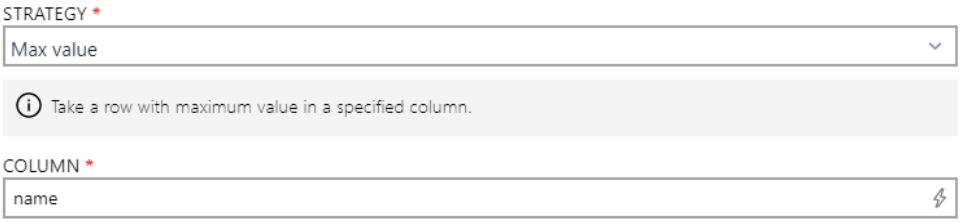
The deduplication step has the following syntax:

```
{
   "id": 11,
   "name": "Deduplicate table Dublicates, take random",
   "type": "deduplicate",
   "source": "Duplicates",
   "target": "Normalized_Duplicates_Random",
   "by": [ "cn", "dn" ],
   "strategy": "random"
}
```

Configure Deduplicate in RayVentory Data Hub



- **SOURCE TABLE**: Enter the table that is used as source for the data. It supports auto-completion and will offer all available tables matching the current input string for selection.
- **GROUP COLUMNS**: Enter a comma separated list of columns which will be used to group the rows.
- **STRATEGY**: Select the strategy that will be used. The following strategies are available.
  - **None**
  - **Random**
  - **Any**
  - **All**
  - **Max value**
  - **Min value**



When **Max value** or **Min value** are defined as **STRATEGY** there will be an additional field available. In the **COLUMN** field, a column from which the value is used needs to be selected.

Split

Splitting is a process of tanking an input table with N row and producing N + X row (X >= 0) by splitting a value from a specific column.

Basic syntax for splitting:

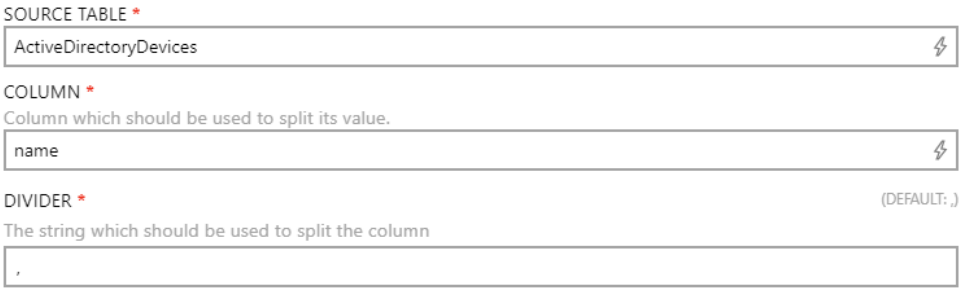
```
{
   "id": 1,
   "name": "<name>",
   "type": "split",
   "source": "<source>",
   "target": "<target_if_persisted>",
   "column": "<the-column-to-split>"
},
```

Source is by default optional, meaning that the table does not have to be existing. If the table is missing, the step will not be executed and its target table will not be written. There is a way to define that the source is required, in which case in case of a missing

```
{
   "id": 1,
   "name": "Unpivot cars table",
   "type": "split",
   "source": "Cars_Pivot",
   "target": "Cars_Split",
   "column": "cars",
   "split": { "type": "vertical", "separator": ";" }
}
```

The separator can have more than one character. Bear in mind, that white-spaces are not trimmed.

Configure Split in RayVentory Data Hub



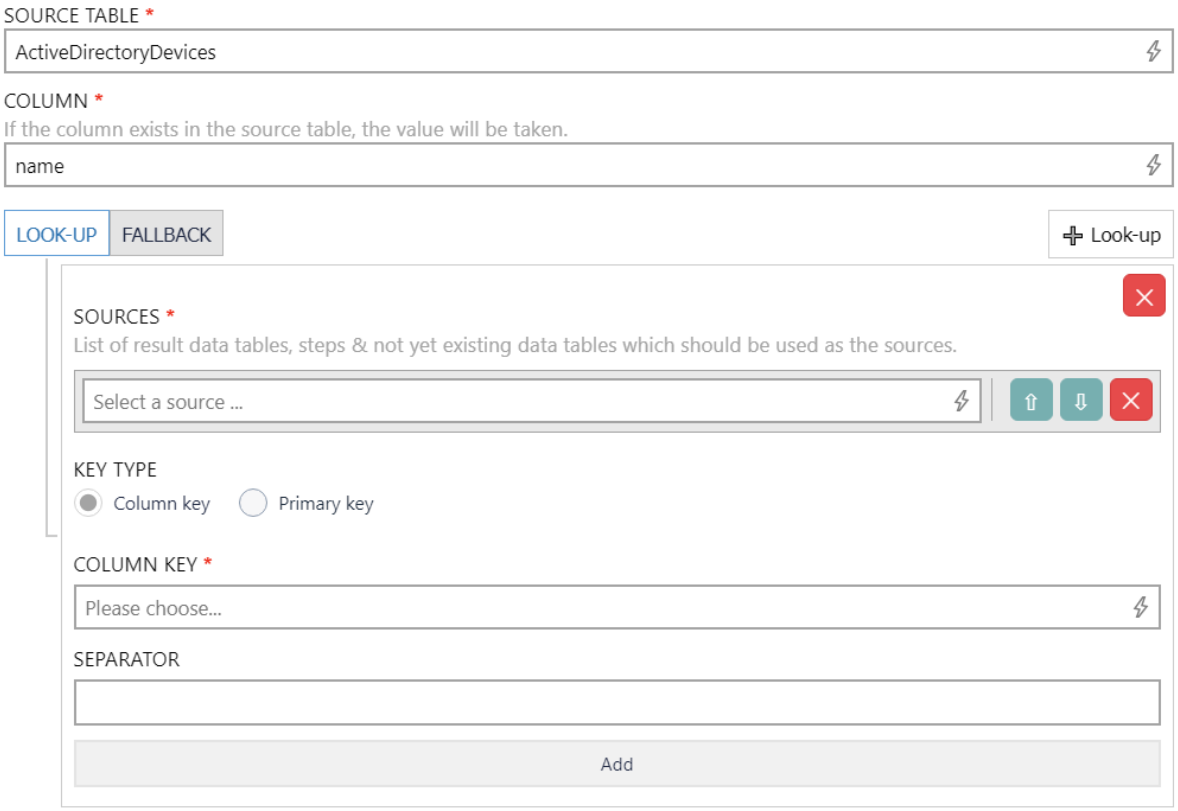
- **SOURCE TABLE**: Enter the table that is used as source for the data. It supports auto-completion and will offer all available tables matching the current input string.
- **COLUMN**: Enter the column that is used to split the value. It supports auto-completion and will offer all available columns matching the current input string.
- **DIVIDER**: Enter the divider (for example: "," or ";") that is used to split the column into this field. It is possible to use more than one character for the divider.

Enrich

Enriching is an operation which does not translate directly to any SQL query. The basic principle of enrichment relies on look-up, where a selected table (enriched table) is being updated by writing values to one of its columns (enriched column), based on values found in other tables.

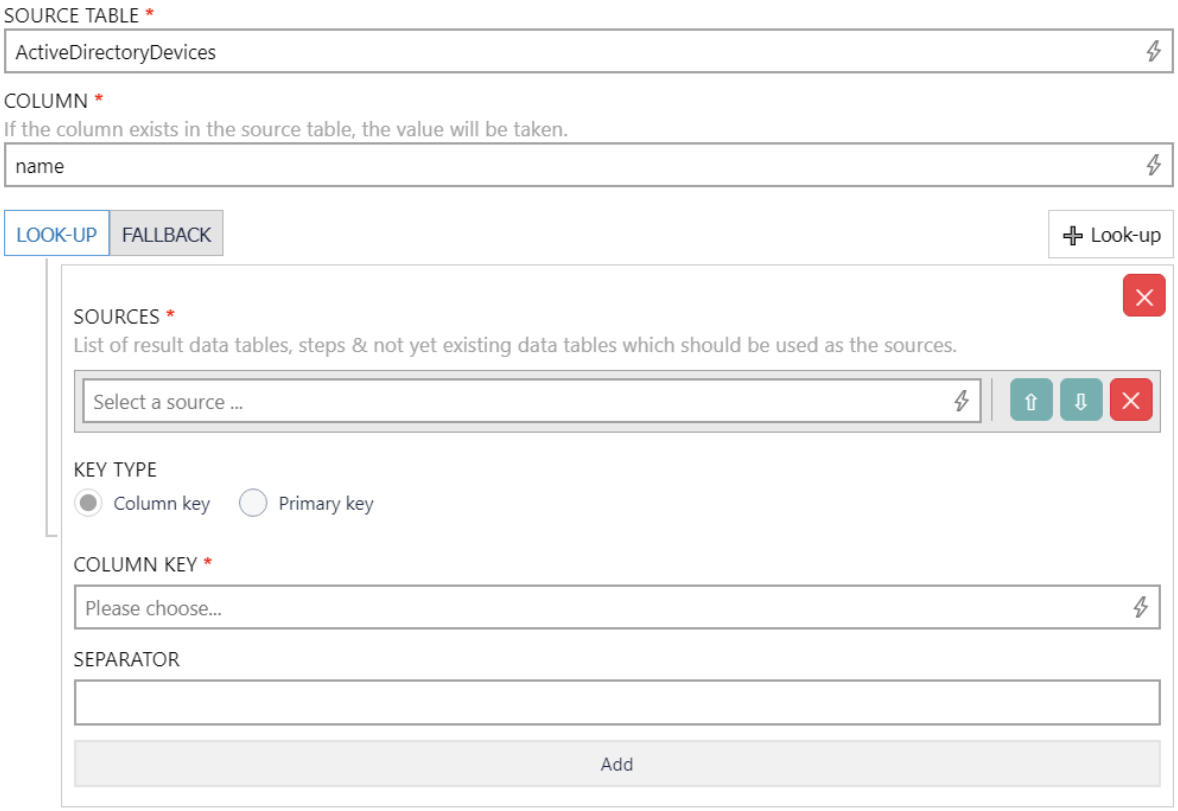
Source is by default optional, meaning that the table does not have to be existing. If the table is missing, the step will not be executed and its target table will not be written. There is a way to define that the source is required, in which case in case of a missing source table the step will fail and report an error. More information about setting up the required steps can be found in the ETL Implementation Guide.

Configure Enrich in RayVentory Data Hub



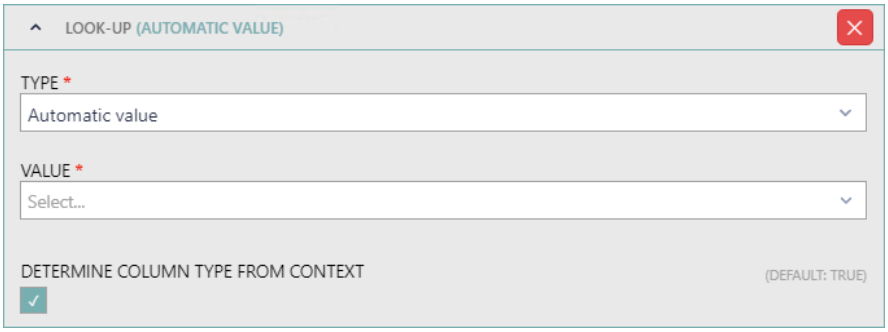
- **SOURCE TABLE**: Enter the table that is used as source for the data. It supports auto-completion and will offer all available tables matching the current input string for selection.
- **COLUMN**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.

LOOK-UP



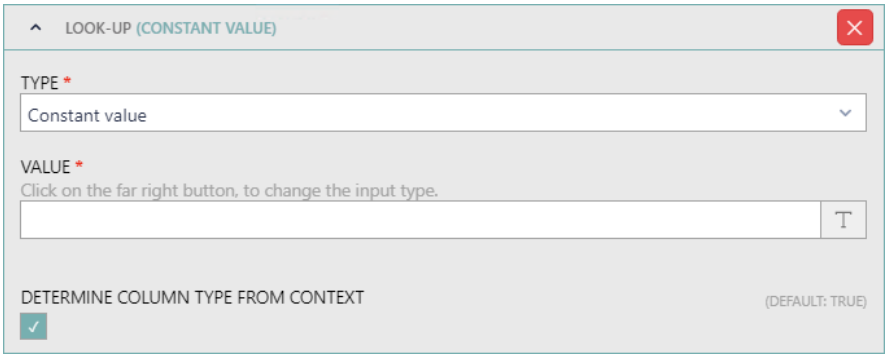
- **SOURCES**: Add a list of resources that should be used as sources. This field supports auto-completion and will offer all available entries matching the current input string for selection.
- **KEY TYPE**: There are two options to choose from for the type of key that will be used.
  - **Column key**: If this option is selected, the COLUMN KEY field will be available. Select the column that will be used as key. This field supports auto-completion and will offer all available entries matching the current input string for selection.
  - **Primary key**: If this option is selected, two field will be available. The PRIMARY KEY field and the FOREIGN KEY field. Select the respective columns for both fields. These fields support auto-completion and will offer all available entries matching the current input string for selection.
- **FOREIGN KEY**: Select a column that will be used as foreign key. This field supports auto-completion and will offer all available columns matching the current input string for selection.
- **SEPARATOR**: A separator can be entered into this field. This field is optional.
- **Add**: The **Add** button can be used to add an additional look-up.

Automatic value



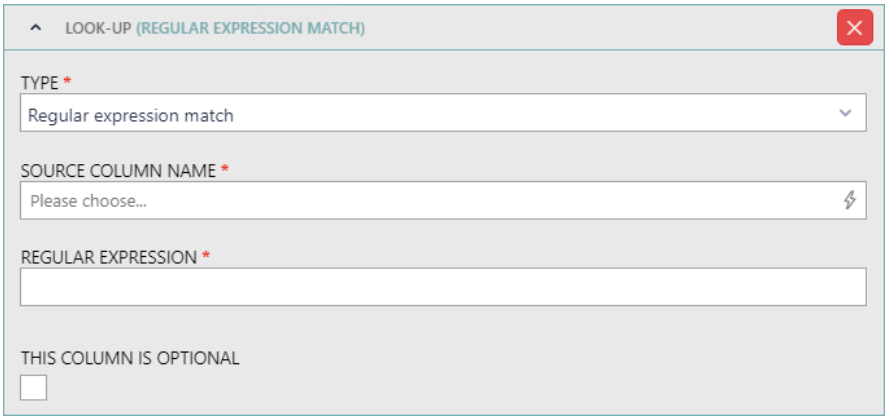
- **TYPE**: Select **Automatic value** to define that an automatic value will be used to determine the value. If **Automatic value** is selected as type, the following fields will be available.
- **VALUE**: Use the dropdown menu to select the automatic value that should be used. The following values are available:
  - **Current datetime**
  - **GUID**
  - **Auto incremented number**
  - **Table name**
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

Constant value



- **TYPE**: Select **Constant value** to define a constant value that will be used to determine the value. If **Constant value** is selected as type, the following fields will be available.
- **VALUE**: Enter the value that should be used. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

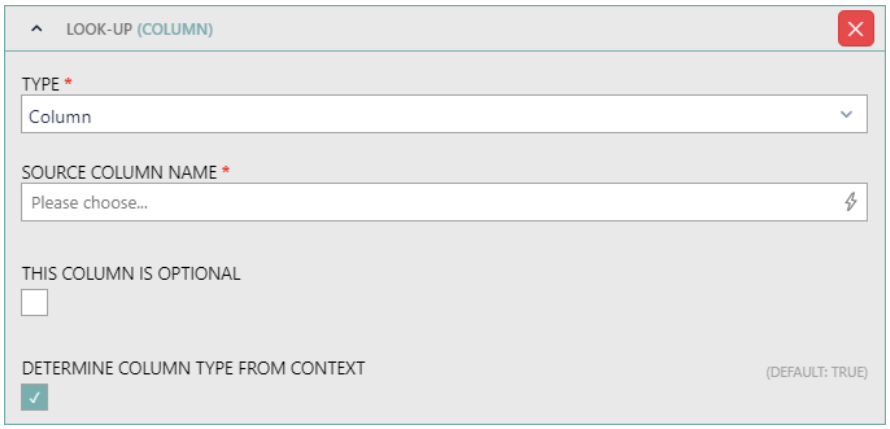
Regular expression match



- **TYPE**: Select **Regular expression match** to define a regular expression that will be used to determine the value. If **Regular expression match** is selected as type, the following fields will be available.
- **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.

- **REGULAR EXPRESSION**: Enter the regular expression that is to be used for the transformation. Detailed information on how to use regular expressions in ETL can be found in the *Programmability* chapter in the *ETL Implementation Guide*.
- **THIS COLUMN IS OPTIONAL**: If this option is not checked, the ETL engine will check if the source column exists before starting the execution. If the option is checked, the engine will start the execution even if the column does not exist.

Column

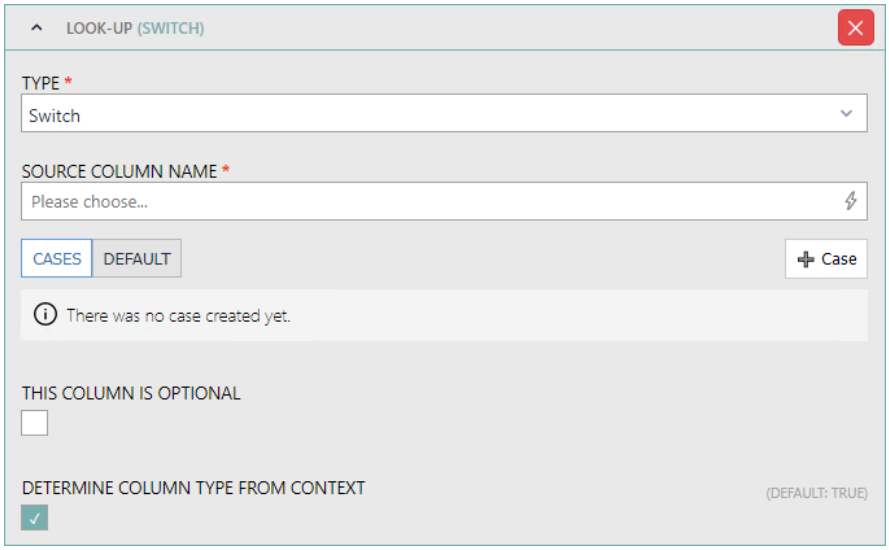


- **TYPE**: Select **Column** to define which column will be used to determine the value. If **Column** is selected as type, the following fields will be available.
- **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **THIS COLUMN IS OPTIONAL**: If this option is not checked, the ETL engine will check if the source column exists before starting the execution. If the option is checked, the engine will start the execution even if the column does not exist.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

SQL

- **TYPE**: Select **SQL** to use an SQL macro to determine the value. If **SQL** is selected as type, the following fields will be available.
- **SQL**: Enter the SQL macro that will be used to determine the value into this field.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

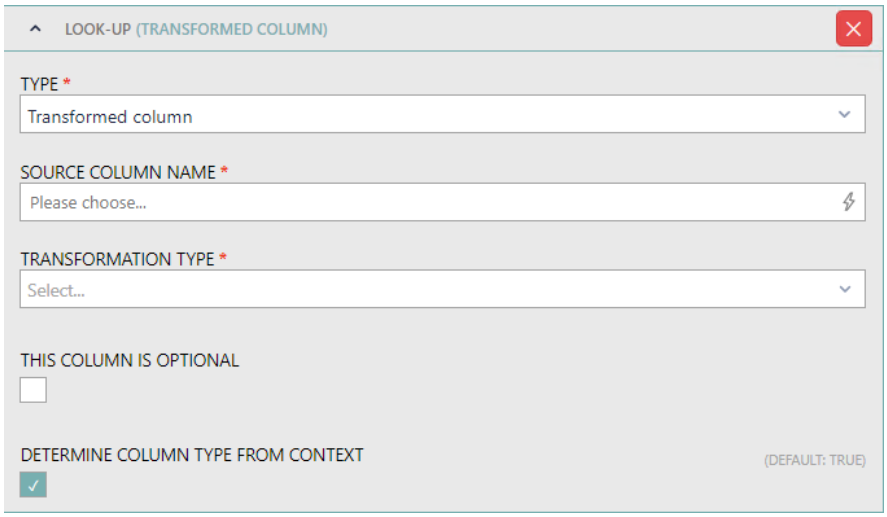
Switch



- **TYPE**: Select **Switch** to use a switch to determine the value. If **Switch** is selected as type, the following fields will be available.

- **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **CASES**: It is possible to add one or more cases. If a case is added the following fields will be available for configuration:
  - **CASE**: Enter a value that should be used for the case. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
  - **THEN**: Enter a value that should be used as then. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
- **DEFAULT**: It is possible to add one default value by choosing default. Then a new field will be available where the value can be defined.
  - **DEFAULT VALUE**: Enter a value that should be used default. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
- **THIS COLUMN IS OPTIONAL**: If this option is not checked, the ETL engine will check if the source column exists before starting the execution. If the option is checked, the engine will start the execution even if the column does not exist.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

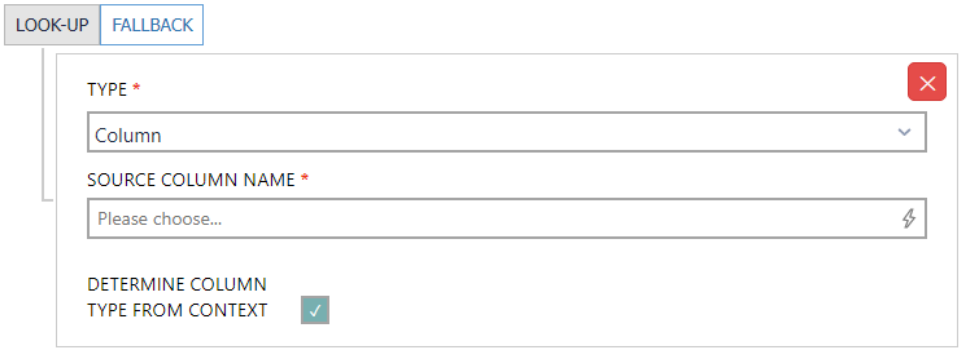
Transformed column



- **TYPE**: Select **Transformed column** to define a transformation type that will be used on the source column. If **Transformed column** is selected as type, the following fields will be available.
- **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
- **TRANSFORMATION TYPE**: Choose one transformation type from the dropdown menu. The following transformation types are available:
  - **Is not null or empty**
  - **Is null or empty**
  - **Length**
  - **Lowercase**
  - **Uppercase**
- **THIS COLUMN IS OPTIONAL**: If this option is not checked, the ETL engine will check if the source column exists before starting the execution. If the option is checked, the engine will start the execution even if the column does not exist.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

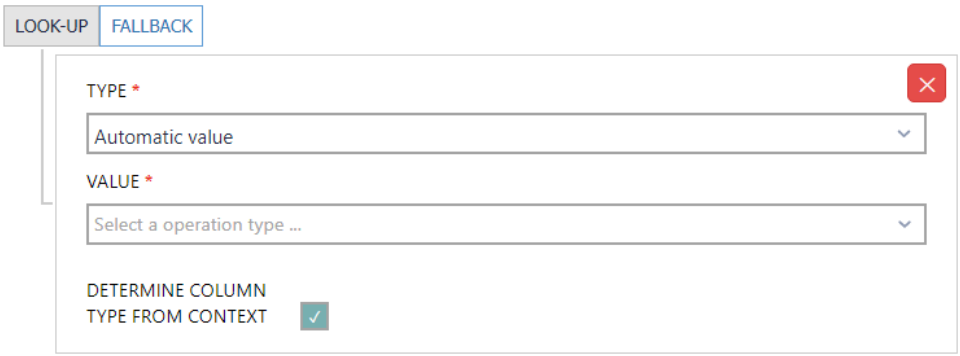
FALLBACK

Column



- **TYPE**: Select **Column** to define the column that will be used to determine the value. If **Column** is selected as type, the following fields will be available.
  - **SOURCE COLUMN NAME**: Enter the column from which the value will be taken. It supports auto-completion and will offer all available columns matching the current input string for selection.
  - **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

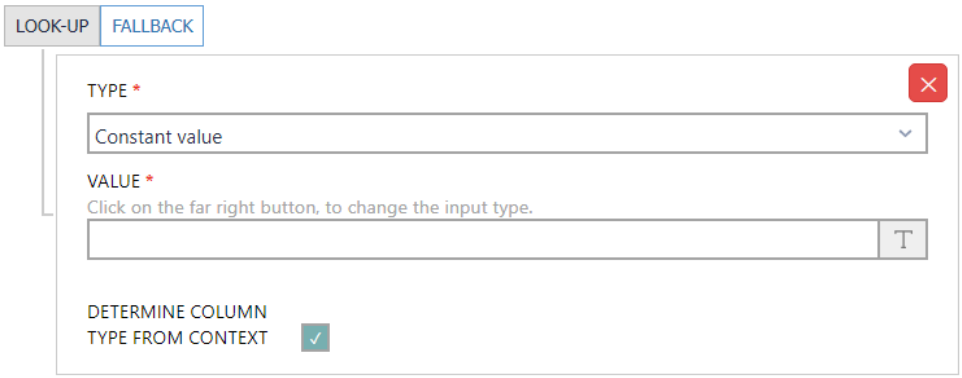
Automatic value



- **TYPE**: Select **Automatic value** to define that an automatic value will be used to determine the value. If **Automatic value** is selected as type, the following fields will be available.
- **VALUE**: Use the dropdown menu to select the automatic value that should be used. The following values are available:
  - **Current datetime**
  - **GUID**
  - **Auto incremented number**
  - **Table name**
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the

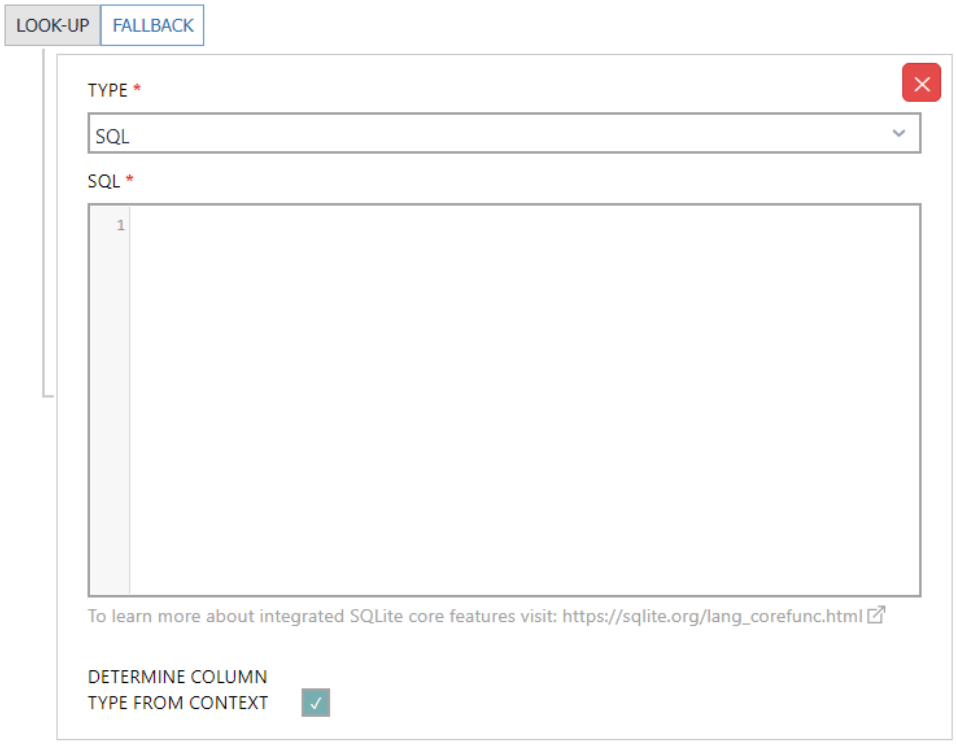
COLUMN TYPE field.

Constant value



- **TYPE**: Select **Constant value** to define a value that will be used. If **Constant value** is selected as type, the following fields will be available.
  - **VALUE**: Enter the value that should be used. By clicking on the button located on the right hand side of the field it is possible to switch the input type between text input, number input, and checkbox input.
  - **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

SQL

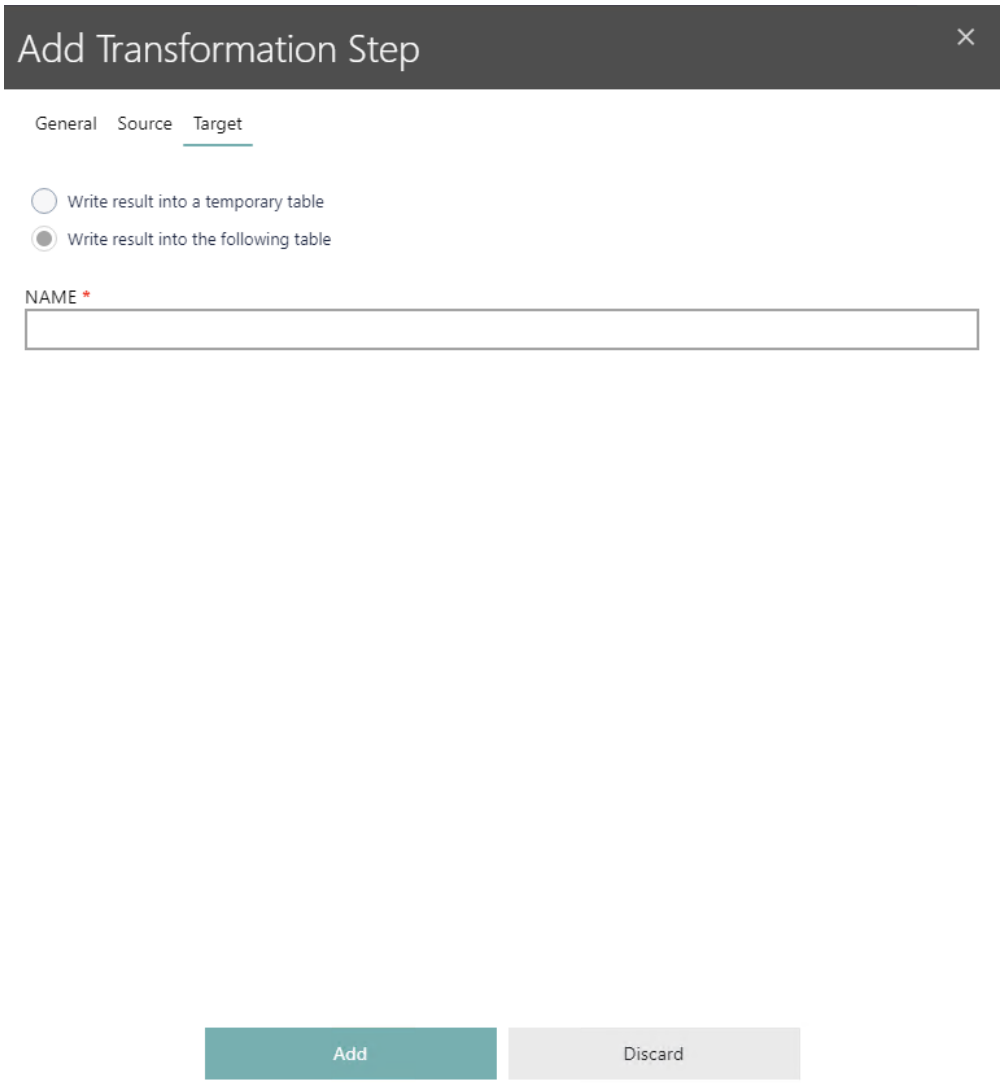


- **TYPE**: Select **SQL** to use an SQL macro. If **SQL** is selected as type, the following fields will be available.

- **SQL**: Enter the SQL macro that will be used to determine the value into this field. Detailed information on how to use scripts in ETL can be found in the *Programmability* chapter in the *ETL Implementation Guide*.
- **DETERMINE COLUMN TYPE FROM CONTEXT**: Enable or disable the automatic determination of the column type. When disabled, the column type must be entered manually into the **COLUMN TYPE** field.

Target

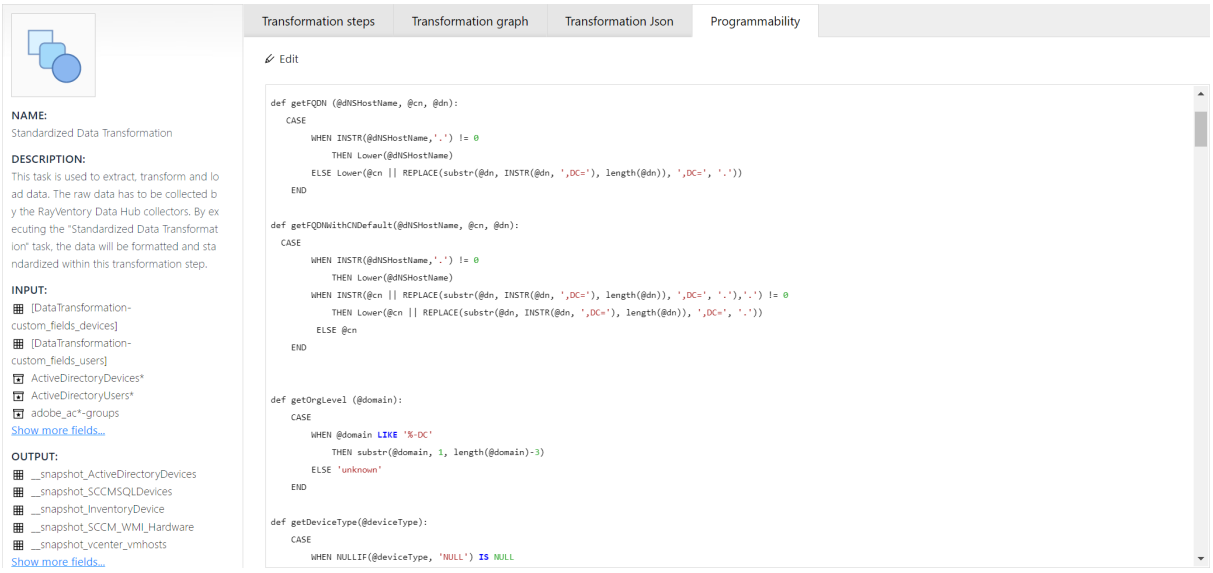
The **Target** tab is used to define the output location for the transformed data.



- **Write result into a temporary table**: Select this option in order to save the data in a temporary table.
- **Write result into the following table**: If this option is chosen, the data will be saved to a specified table.
  - **NAME**: This is the name used for the table. This field is mandatory for the **Write result into the following table** option.

Adding Custom Macros to Transformations

In the **Programmability** tab it is possible to add custom macros to the transformation. Macros can be added or edited by clicking on the **Edit** button located directly above the content area of the tab.



After clicking on the **Edit** button, the **Edit SQL macros** dialog will be opened.

Enter the SQL macros that should be used into the dialog.

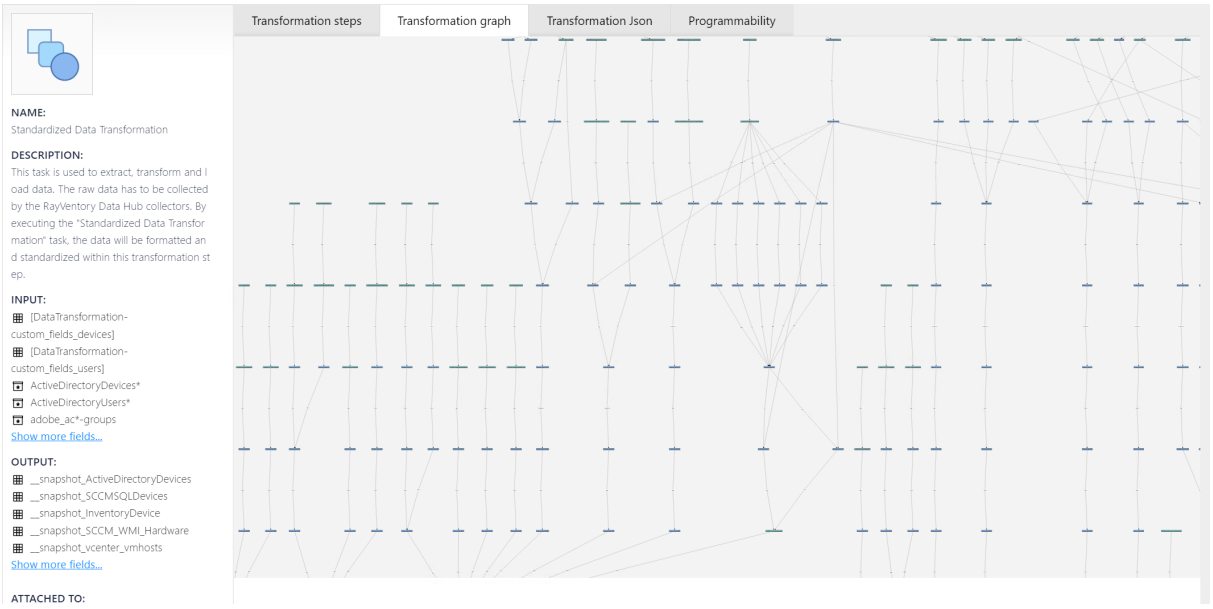
Be aware:
It is necessary to add either a blank or tab at the start of each row below the definition of a macro.

Click on the **Save** button to save the changes and close the dialog.

Detailed information on how to use scripts in ETL can be found in the *Programmability* chapter in the *ETL Implementation Guide*.

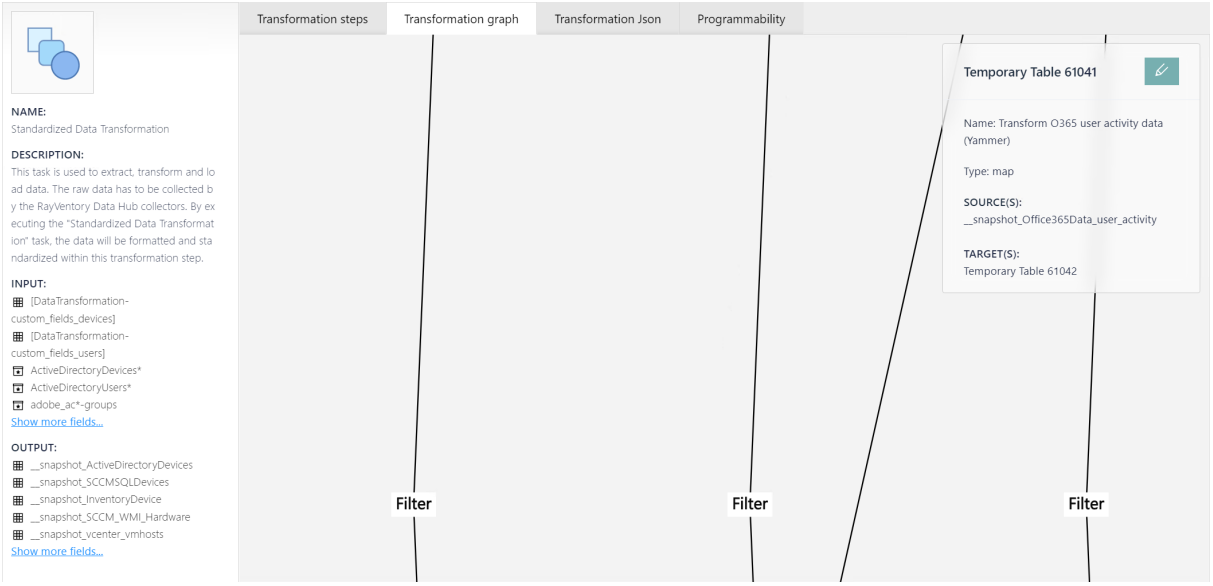
Transformation Graph

In the **Transformation Graph** tab, the selected transformation is shown as a graph. This offers a graphical overview over the different transformation steps and the order they will be executed. Furthermore it shows which steps in the transformation are mutually exclusive.



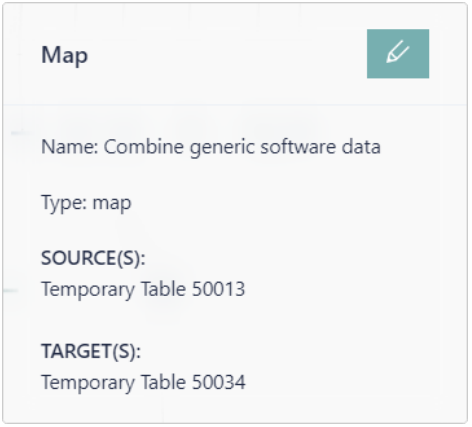
On the left part of the screen general information about the selected transformation can be found. This information include the name of the transformation, a description, as well as the input tables, the output tables, and the tasks which are attached to the transformation.

The transformation graph shown in the right part of the window is calculated once the tab is being opened. It shows a graph of all the steps that are part of the transformation.



To make it even more comprehensive, it is possible to drag the single nodes to another position. This will not change the position of the node in the tree itself, but only the arrangement in the UI where it is shown.

Furthermore, it is possible to click on a node to get more information about that specific node. The information about the selected node will be shown in a small dialog at the top right of the **Transformation Graph** tab.



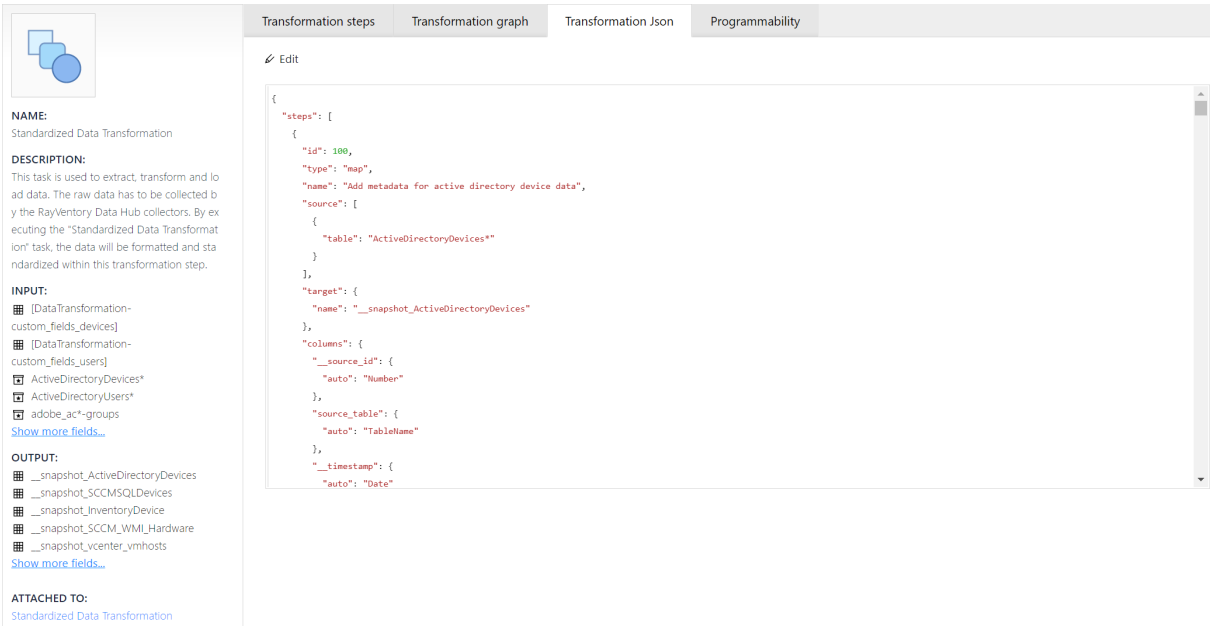
The following information are shown in the dialog:

- **Name**: This field contains the name of the transformation step.
- **Type**: This field contains the type of the transformation step. The type is also shown as header.
- **SOURCE(S)**: This field contains the source table(s) which are used by the transformation step.
- **TARGET(S)**: This field contains the output table(s) which contain the result of the transformation step.

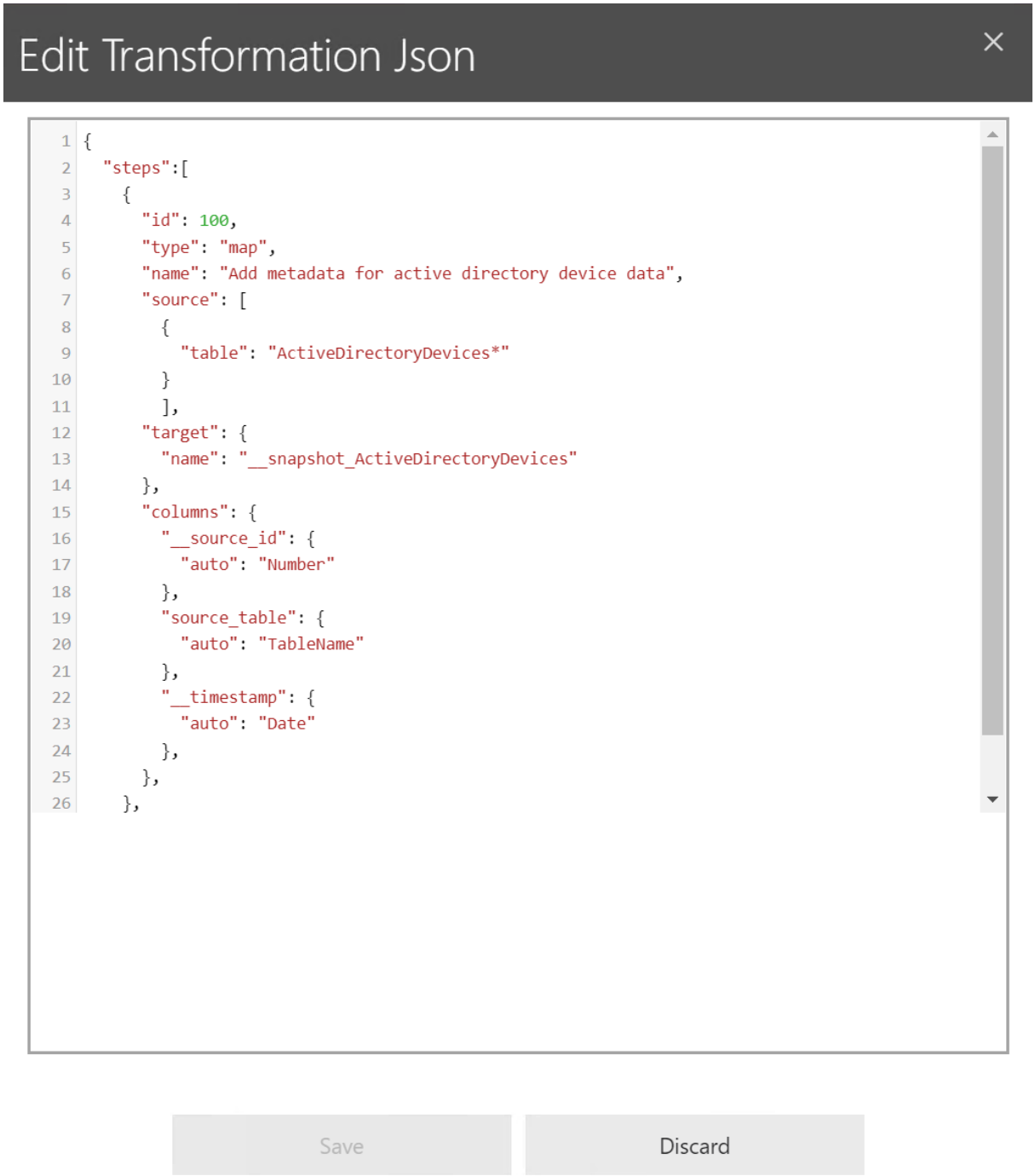
Furthermore, there is an **Edit** button on the top right which can be used to open the **Edit** dialog.

Transformation Json

In the **Transformation Json** tab, the content of the `.json` file that is connected to the selected transformation is shown.



By clicking on the **Edit** button, the **Edit Transformation Json** dialog will be opened.

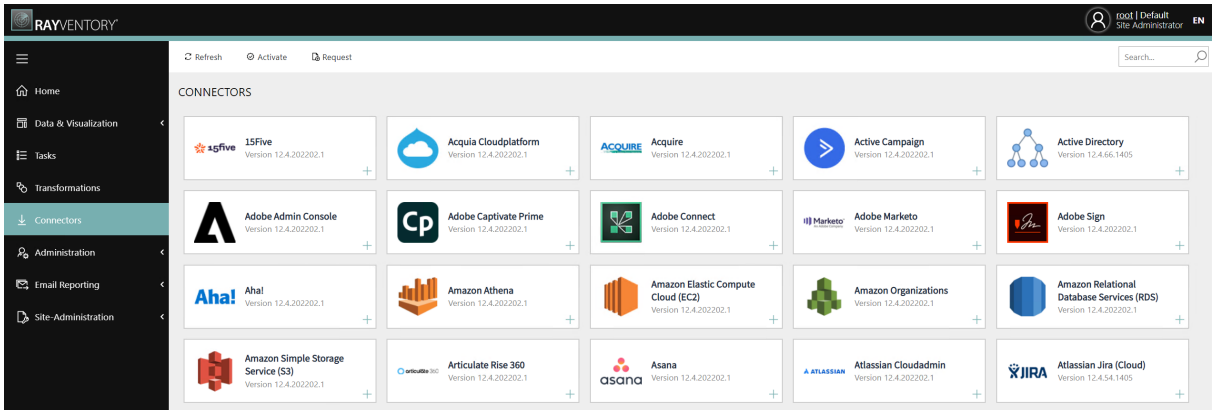


In this dialog the content of the JSON file can be added directly. After completing all changes, click on the **Save** button. As soon as the changes have been saved to the file, they will also be visible in the **Transformation steps** and the **Transformation graph** tab.

Connectors

The **Connectors** page is the central place where external connectors can be managed and licensed. A connector is a self-contained piece of software which is specialized in getting data

from one or more external systems.



The list is sorted alphabetically, and shows the name, logo and the internal version of the connector. It is possible to use a filtering mechanism, which is provided in the top right corner.

A connector may have one of the following states:

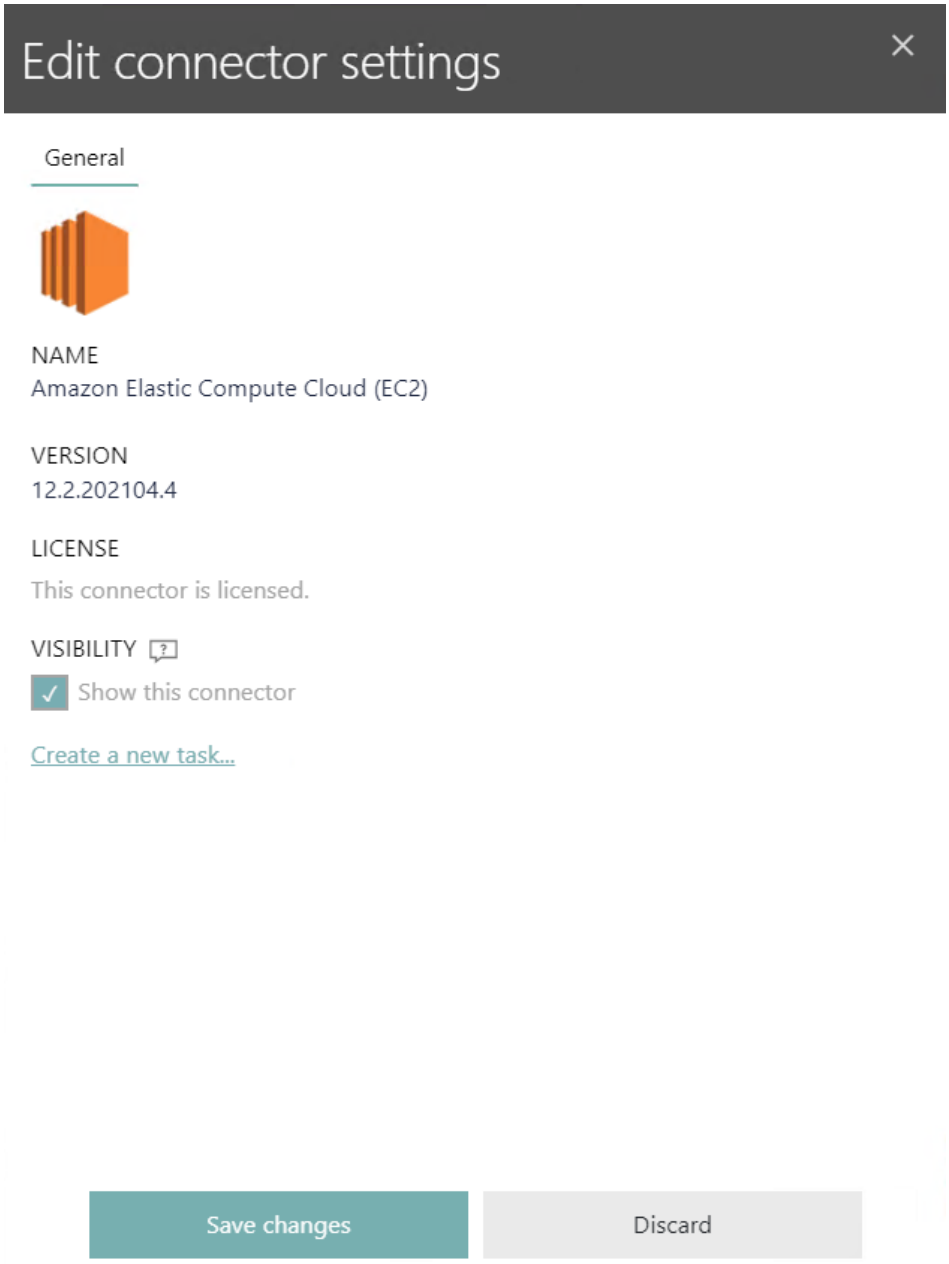
- **Activated and visible**
  The connector is ready to be used.

- **Activated but hidden**
  The connector is covered by the current license, however it has been hidden. This may be the case if the given platform that the connector supports is not actively used, and the connector is hidden to not clutter the list of available target systems when configuring a task.

- **Not activated and hidden**
  The connector has not been activated, and it is hidden. This is the default for clean installation.

- **Not activated but visible**
  It is possible to make the connector visible although no license is available. This may be the case if the license will be provided later, after initial set-up of the data, reports and tasks. Note that before the connector can be actually used (running a task), a valid license will be required.

In order to use a connector, it must be activated first. After a clean install, most of the connectors are hidden by default. To show a connector, click it with left mouse button and tick the respective checkbox:



Once the connector is active, it can be used to define tasks. It is not possible to run tasks or accept incoming data, if the underlying connector is not licensed. To proceed, a valid license needs to be provided, as described in the next chapter.

Creating a New Task from the Connectors Page

You can create a new task directly from the Connectors page. To do it, press the plus icon next to the connector tile. You will be redirected to the **Tasks** page, with proper type preselected for you. Detailed information on the separate connectors available for RayVentory Data Hub can be

found in the *RayVentory Data Hub Connectors* document.

Activating Connectors

To activate one or more connectors, a license key provided by Raynet is required. License key for a connector is not the same as your product key, as they both are licensed using different license models.

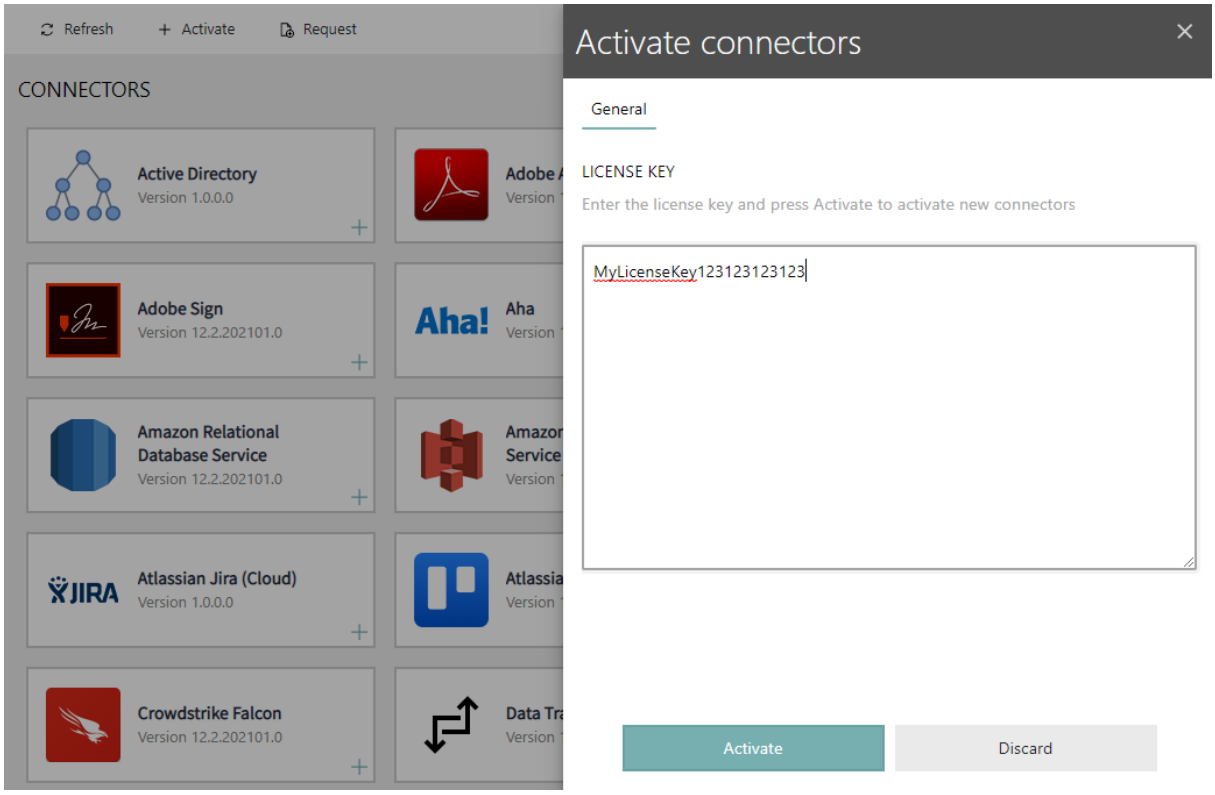
The following connectors are always licensed and require no extra license key:

- Active Directory
- Data Transformation
- Microsoft SQL Server
- ODBC
- PowerShell
- RayManageSoft UEM
- RayVentory Catalog
- RMS Stored Procedures
- SCCM (via WMI)
- VMware vCenter

License keys embed various information, including the list of activated connectors, user data and expiration date. You can have several activated license keys, as they are cumulative.

Note:
Activation of connectors is always done on tenant basis.

To open the activation dialog, go to the **Connectors** page, and press **Activate** from the top bar.



Provide the license key, making sure it is pasted as-is, without adding white spaces or trimming non-white space characters.
After pressing the **Activate** button, the license key will be verified, and the connectors will be automatically activated.

If a newly activated connector has not been set as visible before, it will be automatically made visible.

Administration

This area is restricted to **administrators** only.

It allows you to perform the following administrative tasks:

- Managing project users (note: only users scoped to the current tenant are visible and manageable)
- Managing variables
- Managing user groups (only tenant-specific)
- Viewing and modifying advanced tenant settings
- Downloading MSI for installation of agents.

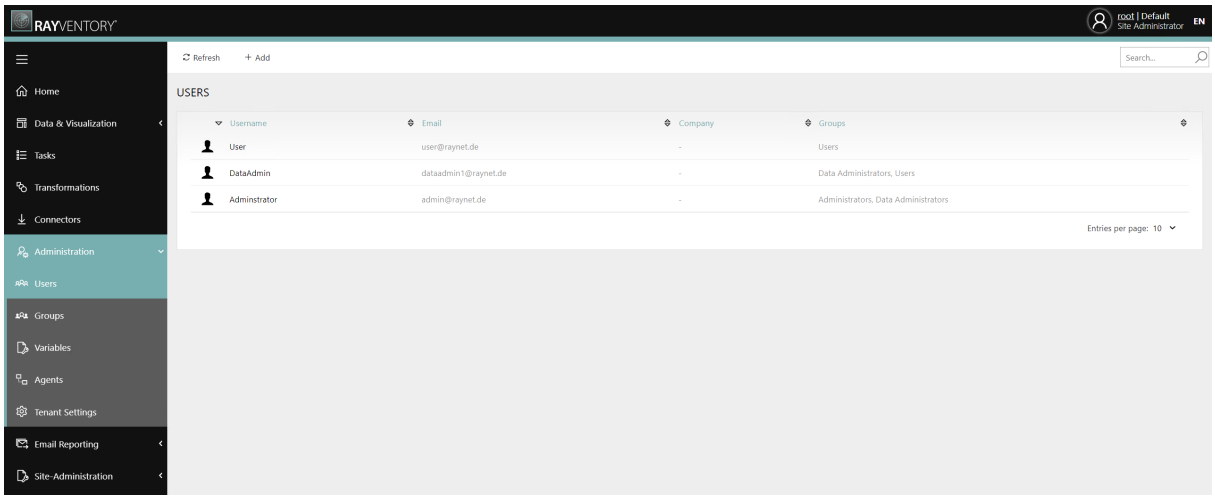
Users

User management is available only for administrators. Navigating to the user's view through the navigation bar shows all users in a table. The action bar has the following functions:

- **Refresh**
  Reloads all users and updates the table.

- **Add**
  Opens the "**Add User**" panel that allows it to create a new user.

On the right side is the table filter, which allows you to filter all users by name.

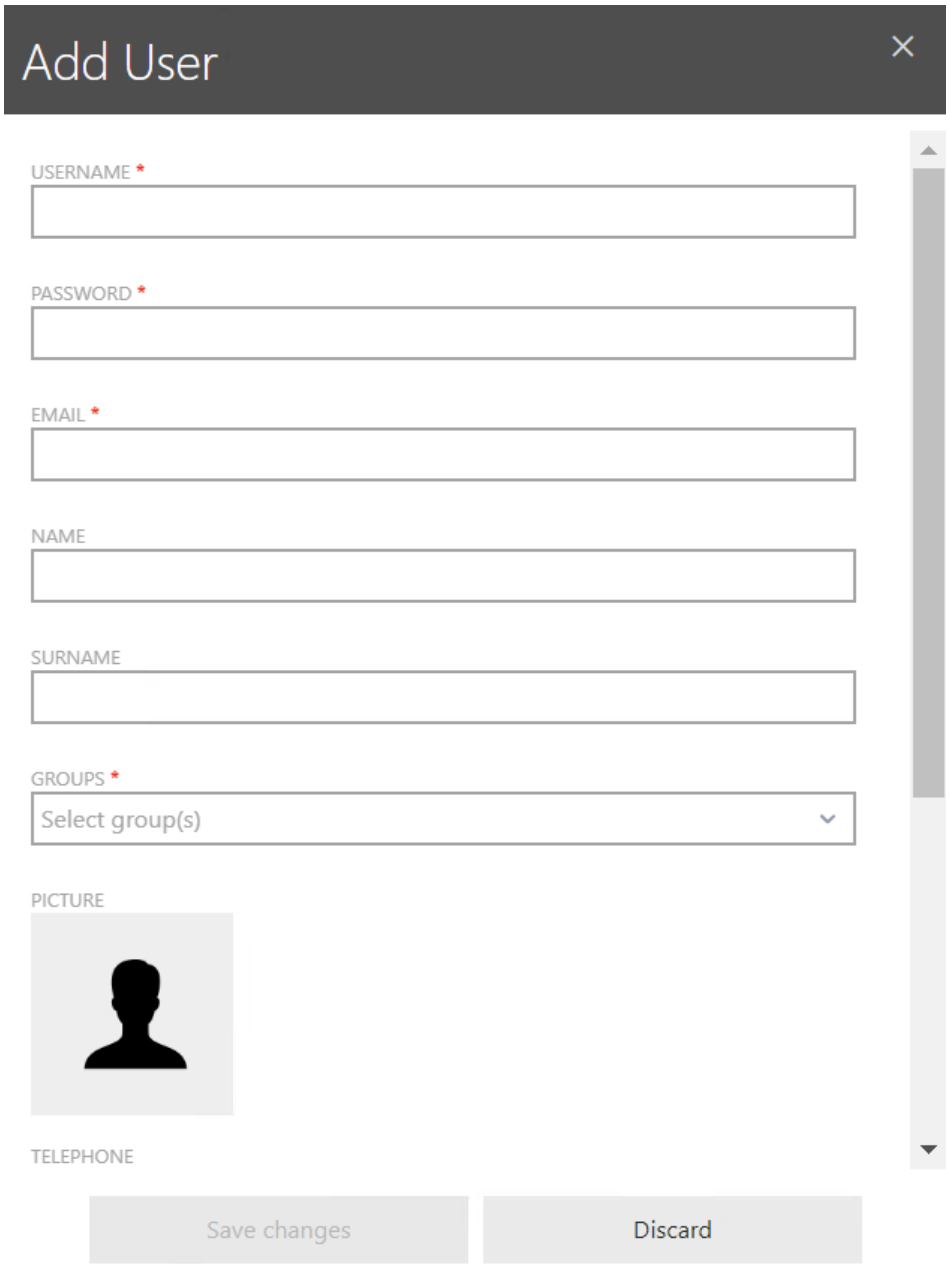


RayVentory Data Hub contains no predefined users after a clean installation except of the root user. Setting up required tenants, users and group assignments is an initial task for which the root account can be initially used, but it is generally not recommended to use root accounts later once the instance is set-up and up and running.

For more information about the default users, see the Default Users chapter.

New User

By clicking on the **Add** button from the actions bar in the users view opens a right side panel showing a form to create a new user.

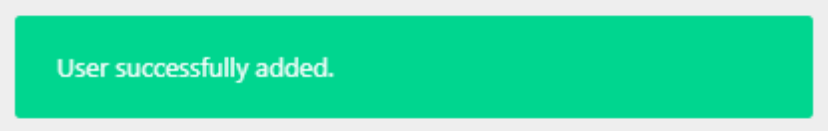


Fill in the required fields:

- **Username** – The name of the user. Name must be unique.
- **Password** – Password of the user.
- **Group** – Group(s) of the user. The group determines the authorization level of the user. Groups and their authorization level are configured in the Groups view.

Optionally, you can add a profile picture of the user.

Confirm the inputs by clicking on the button "Save changes" at the bottom. If the user is successfully created, a green pop-up notification is displayed in the top right corner.



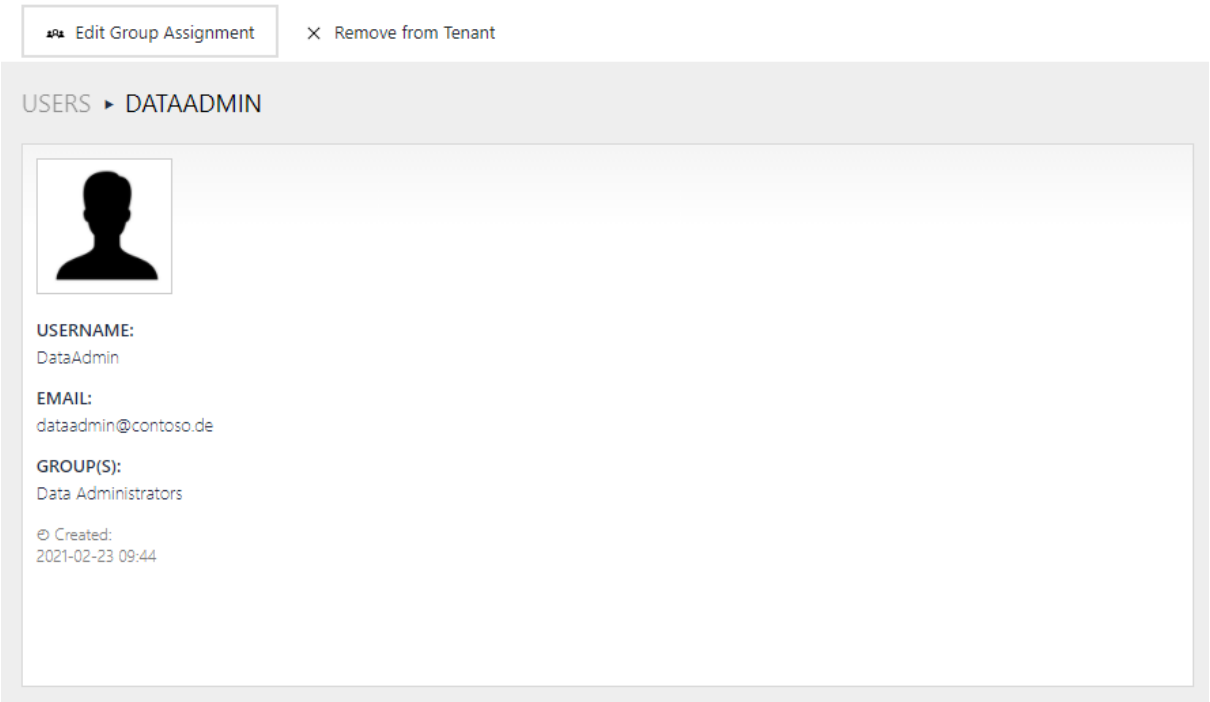
If any error occurs, e.g. not all mandatory fields were filled or a field requires a unique name, a red pop-up notification is shown with further information. Furthermore, any invalid input field is marked with a red border and displays a short error message below the input field.

Closing the editor by clicking either on the “Discard”-button in the bottom right corner or clicking on the cross in the top right corner closes the editor without saving your changes.

User Details

Clicking on a table row in the users view navigates to the user details of the selected user. In this view the user details and the recent changes made on the details of this user are displayed. In addition, the user can be edited and deleted using the action buttons located in the upper horizontal action bar.

- **Edit** - Opens the “Edit User” panel that allows it to edit the current user details. This panel is displayed very similar to the “Add User” panel prefilled with the current user details.
- **Delete** – Deletes the current user. As a consequence, this user can no longer log into the system.



Default users created by the system initially have no history of changes. When creating a new user manually or editing its details, e.g. changing its username or group assignment, will create a new history entry that tracks the time when the action was done, the action (creation or modification) and the user who executed this action. The history of changes can be reviewed by pressing the **Created** / **Last modified** link.

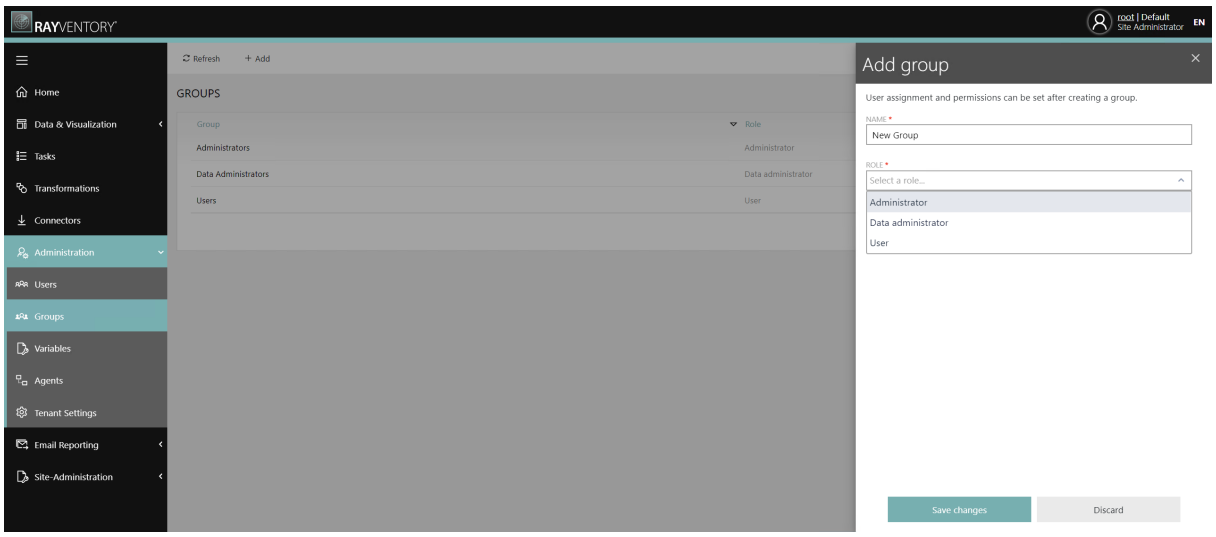
Groups

New Group

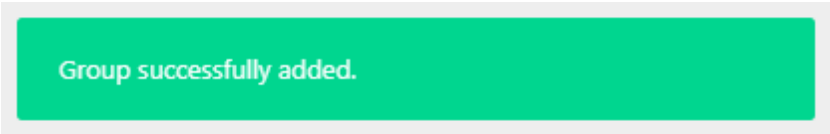
By clicking on the **Add** button from the actions bar in the groups view opens a right side panel showing a form to create a new group.

Fill in the required fields:

- **Name** – The name of the group. Name must be unique.
- **Role** – Specifies which role the user will have when assigned to this group. RayVentory Data Hub introduces three user roles for selection, which are presented in the "User Roles" section.



Confirm the inputs by clicking on the button "Save changes" at the bottom. If the user is successfully created, a green toast notification is displayed in the top right corner.



If any error occurs, e.g. not all mandatory fields were filled or a field requires a unique name, a red toast notification is shown with further information. Furthermore, any invalid input field is marked with a red border and displays a short error message below the input field.

Group Details

Clicking on a table row in the groups view navigates to the group details of the selected group. In this view the group details and a list of users assigned to this group is shown initially. The tab content can be changed by selecting a tab from the tab selection box. A list of permissions on reports and dashboards is shown when selecting the Permissions tab. A list of recent changes made on the details of this group are shown when selecting the Recent Changes-tab. In addition, the group can be edited and deleted using the action buttons located in the upper horizontal action bar.

- **Edit**
  Opens the **Edit User** panel that allows it to edit the current group details. This panel is displayed very similar to the “Add User” panel pre-filled with the current group details.

- **Delete**
  Deletes the current group. Note: Deleting a group is only possible when no user is assigned to it.

Role Permissions

RayVentory Data Hub introduces three user roles and an anonymous access for shared items. Depending on the role of a user, the user has a restricted view or no access to certain pages and functionality.

The following table provides an overview of the view, write and delete permission per view for each role. The list is not comprehensive and shows only a selection of screens and features.

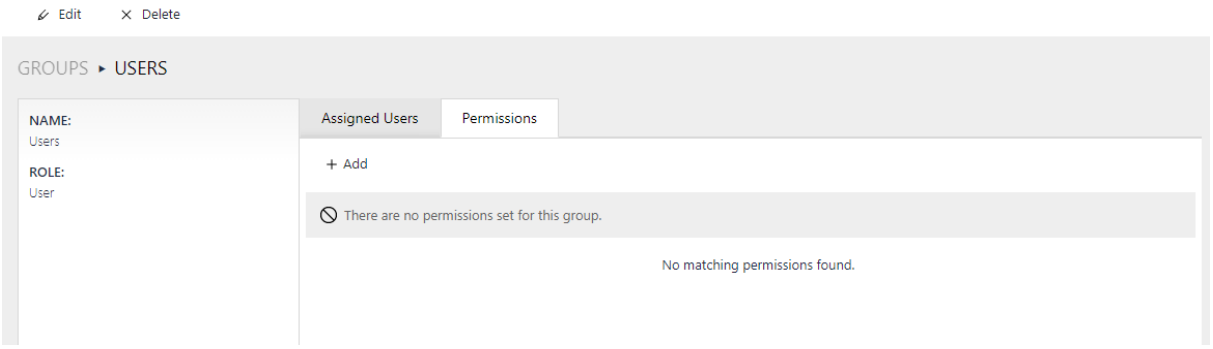
Roles	Login	Tasks			Library Items				
	View	View	Write	Delete	View	Write	Delete	Import	Export
Admin	+	+	+	+	+	+	+	+	+
Data Admin	+	+	+	+	+	+	+	+	+
User	+	-	-	-	(Conditional)	(Conditional)	(Conditional)	-	-
Anonymous	+	-	-	-	(Shared public link)	-	-	-	-

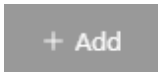
Agents			Agent			Users			Groups			Settings
View	Write	Delete	View	Write	Delete	View	Write	Delete	View	Write	Delete	View
+	+	+	+	+	+	+	+	+	+	+	+	+
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-

Library Item Permissions

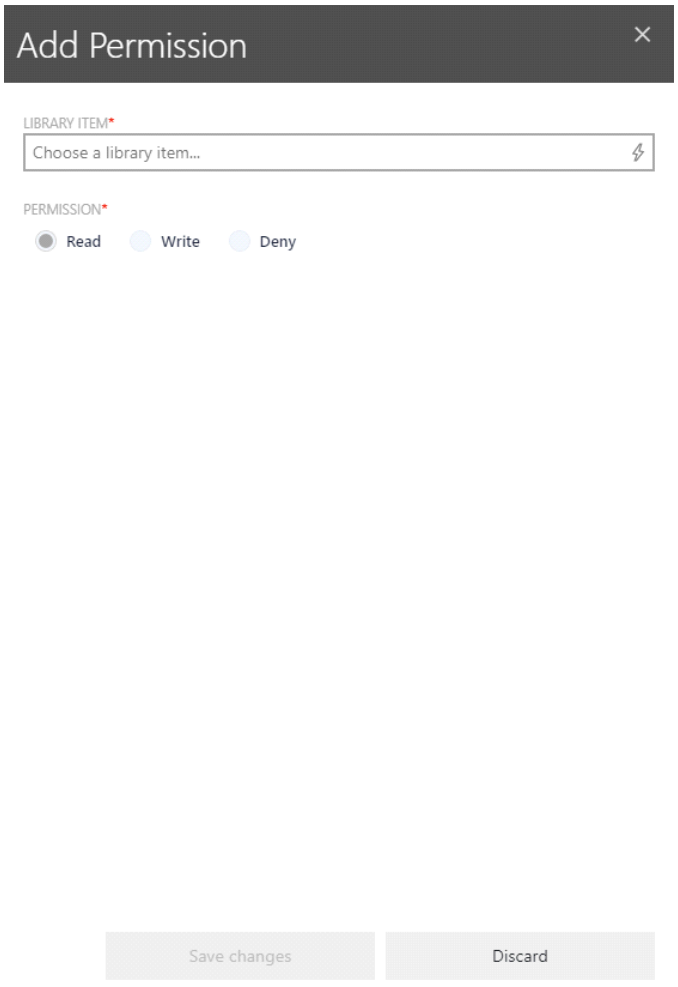
Users that are assigned to groups whose role is either Administrator or Data Administrator always have read and write access to all reports and dashboards. Users who are assigned to a group or groups whose role is User only have read or write access to a report or dashboard if the permissions is explicitly given.

To grant a group of users with the User role access to specific reports and dashboards, first navigate to the group detail view of the group. Note that only the user with an Administrator role is allowed to view and edit groups. Next, open the Permissions-tab by selecting it from the tab selection box.





Click the button to the top left. A side panel opens to the right, showing a form to create a new permission.



Fill in the required fields:

- **Library Item** – Select a library item you want to grant access to. This can be either a Section, Report or Dashboard
- **Permission** – Specify if the users of this group have read or write access. Selecting deny will explicitly deny the access to the selected library item.
- **Inheritance** – When selecting a section, the optional checkbox inheritance is shown. Checking this checkbox implies that all child elements (including sections and their child elements recursively) of the section will inherit the permission selected for that section.

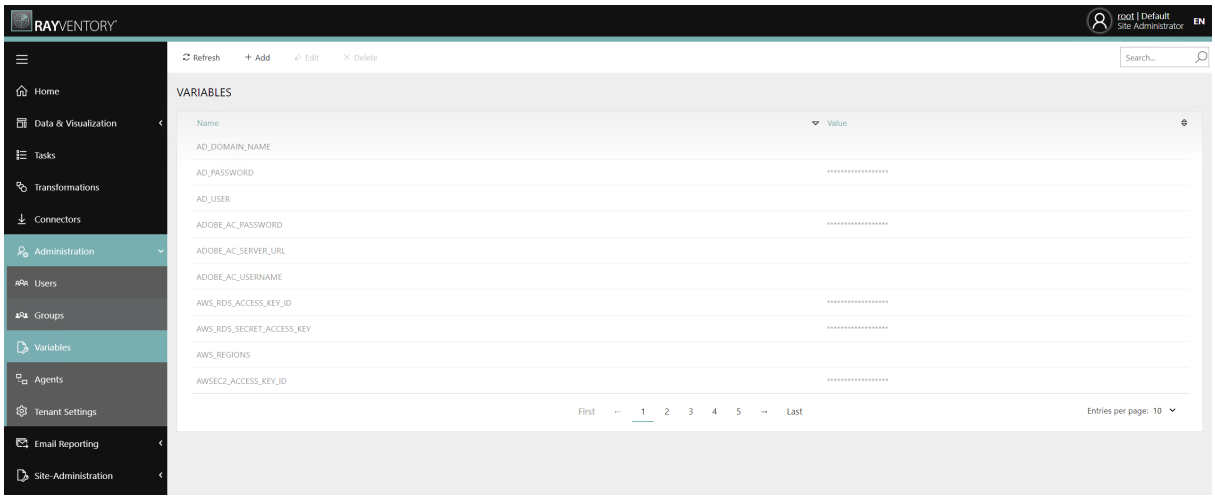
Note:
It is still possible to set specific permissions for the child elements of this section. A permission set explicitly always has priority over the inheritance.

Confirm the inputs by clicking on the button "Save changes" at the bottom. If the user is

successfully created, a green pop-up notification is displayed in the top right corner.
If any error occurs, e.g. not all mandatory fields were filled, a red toast notification is shown with further information. Furthermore, any invalid input field is marked with a red border and displays a short error message below the input field.

Variables

Variables are special tokens which associate a value to a unique key, under which the variable is accessible. Their primary purpose is to abstract configuration details from their tasks. The abstracted properties can include both sensitive data (like credentials, password, tokens and connection strings) and non-sensitive data (public tokens, options, URLs etc.).



A variable can be of one of the following types:

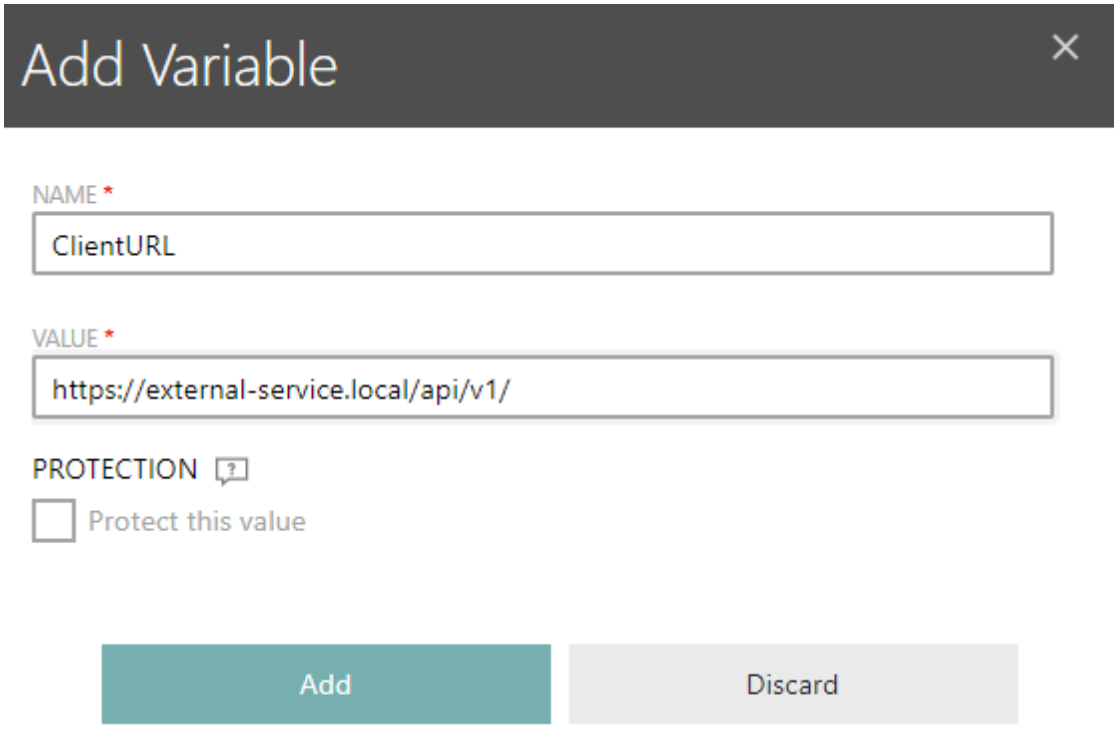
- **Normal variable** - is displayed to the end user in an open-text way, and the user can change its value with a text input. Variables of this types are good use for a data which does not contain sensitive information, including but not limited to credentials.

- **Protected variables** - are always masked with password characters, and the real value is never transmitted to the UI. This type of variables should be used for password, tokens and other credentials, where it is vitally important to hide their content from application users. The users can always change their value, but once the value is saved no one is able to retrieve it back.

Note:
Protected or not, variables of both types are always stored encrypted in the database, using tenant-specific encryption key.

Adding a Variable

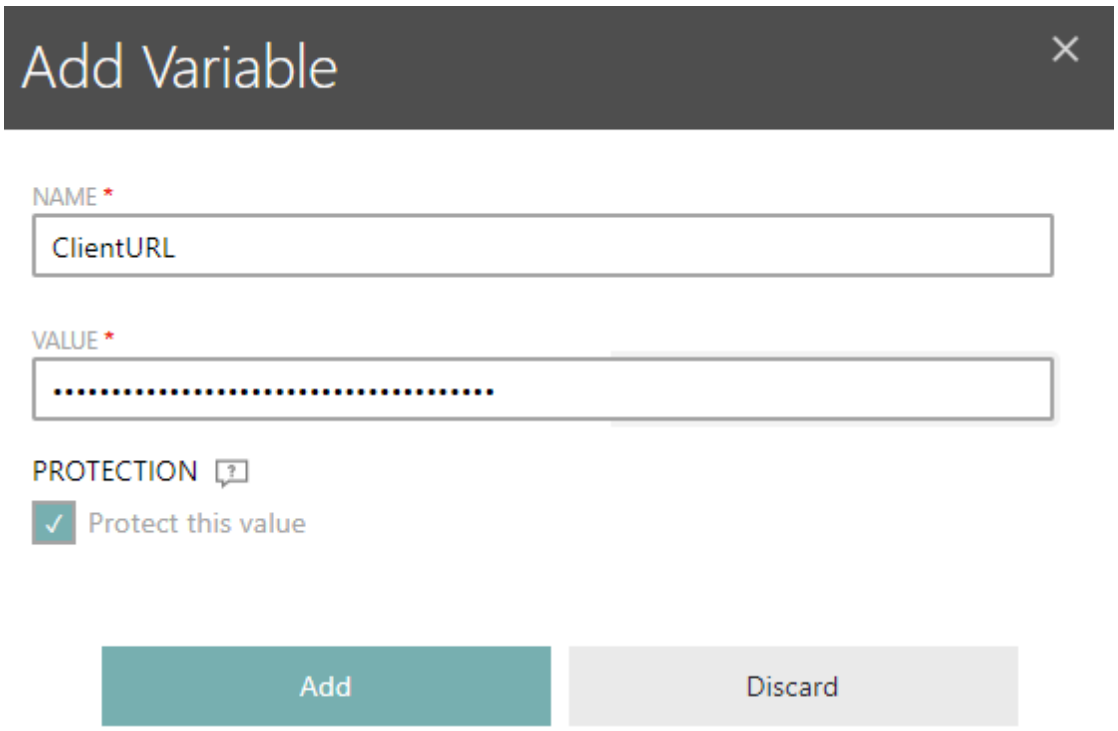
To add a variable, go to the **Variables** page (under **Administration** menu) and press the **Add** button. An edit form will be opened in the sidebar:



Each variable must be identifier with a unique name. Tasks can reference their values, using the following syntax: `###VariableName###`. Except of being unique, variable names must not use special characters. It is also recommended that the name of a variable describes what it is for.

The value may be extra protected (see chapter Variables for more information about what the protection does). By default, a new variable is not protected, which means its value can be entered using a plain text box.

You can make the variable protected by ticking the checkbox **Protect this value**. This will turn the input text box into a password box:



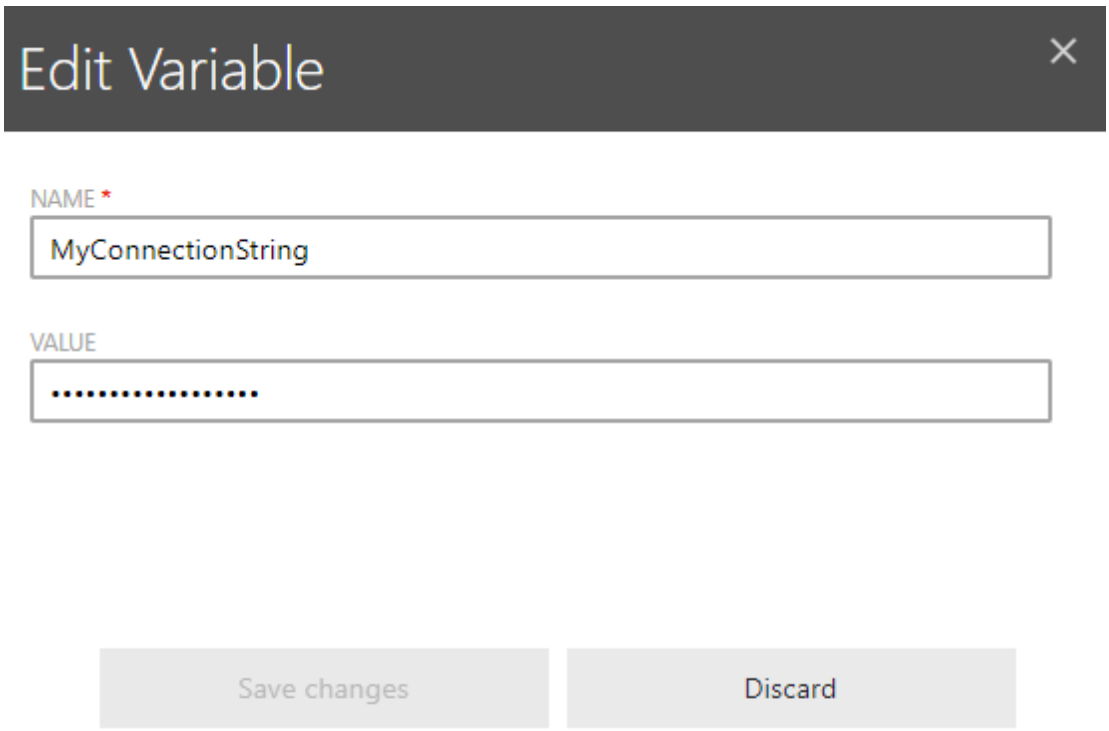
Warning:
After saving changes, it is not possible to change the type of the variable anymore. Make sure to set this up initially, otherwise the only way to have a variable with a different protection level is to drop the old one and create a new one with the same name.

It is recommended to use protected variables for passwords, credentials and other sensitive information.

Once you are ready, press **Add** to add the new variable and close the dialog.

Editing a Variable

To edit a variable, go to the **Variables** page (under **Administration** menu), select the variable to be edited and press the **Edit** button. An edit form will be opened in the sidebar:



Each variable must be identifier with a unique name. Tasks can reference their values, using the following syntax: `###VariableName###`. Except of being unique, variable names must not use special characters. It is also recommended that the name of a variable describes what it is for.

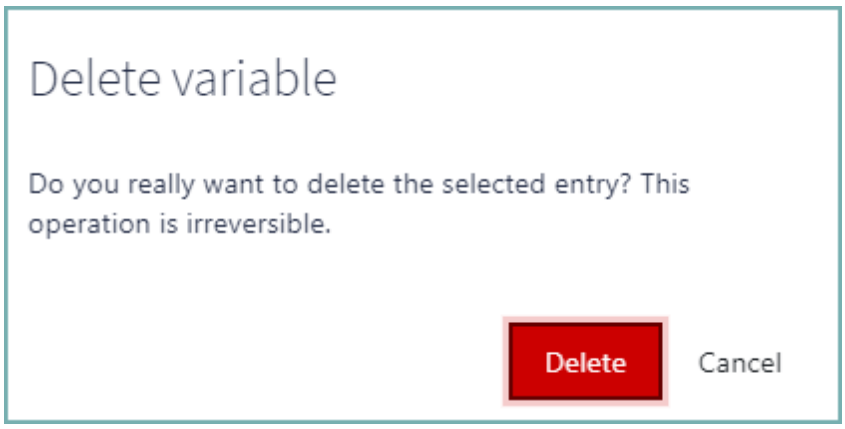
If the variable has been marked as **protected**, its value will be shown in a masked password box (see above). Otherwise, the value can be seen in clear-text.

> **Note:**
> While editing the variable, it is not possible to change its protection type (see chapter Variables for more information about what the protection does). This means, that a conversion from a password variable to a clear-text variable or vice-versa is not possible, and must be done instead by recreating a variable and dropping the existing one.

Once you are ready, press **Save changes** to save changes and close the dialog.

Deleting a Variable

To delete a variable, go to the **Variables** page (under **Administration** menu), select the variable to be deleted and press the **Delete** button. Once confirmed, the variable will be removed.



Removing a variable does not change the tasks that may refer it. They may however stop working, because the value of placeholders will not be resolved once the variable is removed.

Auto Variables

As of version 12.5, RayVentory Data Hub has a set of predefined variables which are always available, even if not defined directly in the **Variables** tab.

- **RESULT_DATABASE_CONNECTION_STRING**
  This resolved to the full connection string of the database, containing the results.

- **RAYVENTORY_CATALOG_URL**
  This resolves to the URL of RayVentory Catalog module. By default the URL is `https://rayventorycatalog.raynet.de`.

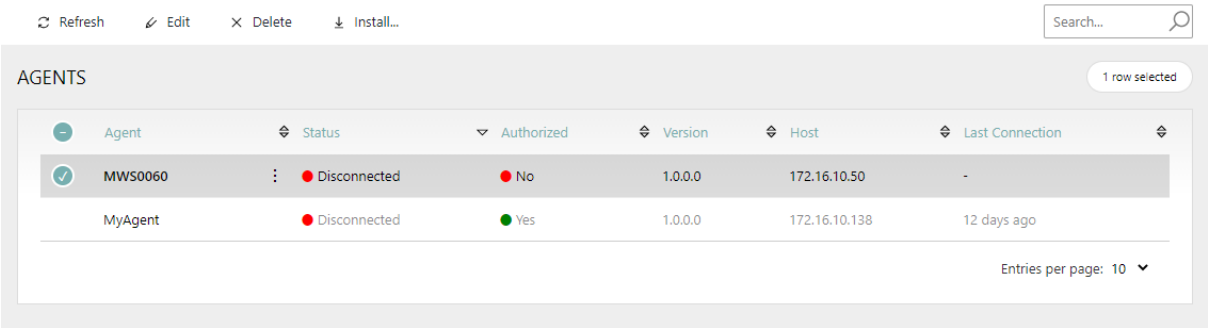
Auto variables are available in drop-down selectors (see chapter Referencing Variables for more information).

Agents

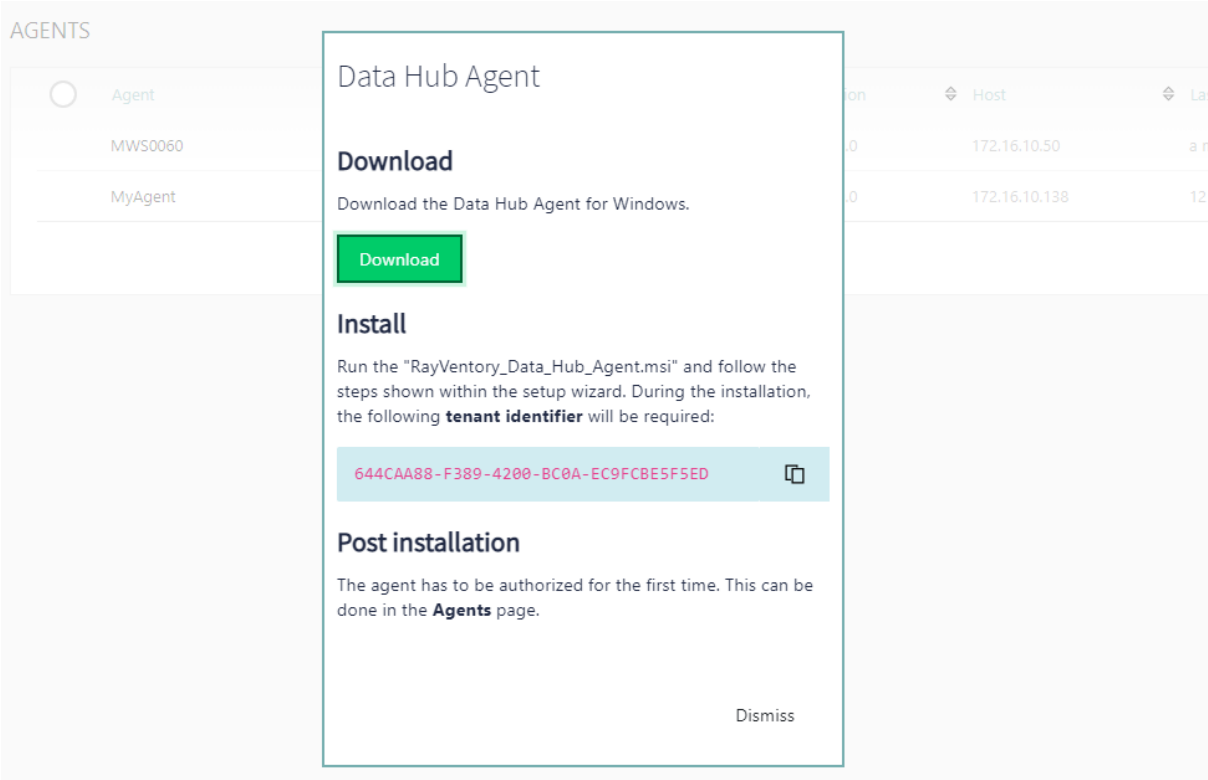
Agents page is a sub-page of the **Administration** section, which let administrators manager, install and authorize agents.

The following contextual actions are available:

- **Refresh** – reloads all agents and updates the table
- **Edit** – opens the edit panel, which allows it to edit a single selected agent. The Edit-button is only enabled when a single agent is selected.
- **Delete** – deletes the selected agents. Deletion requires a confirmation from the user.
- **Install...** - open a dialog where the user can download an agent MSI and guides through the next steps.

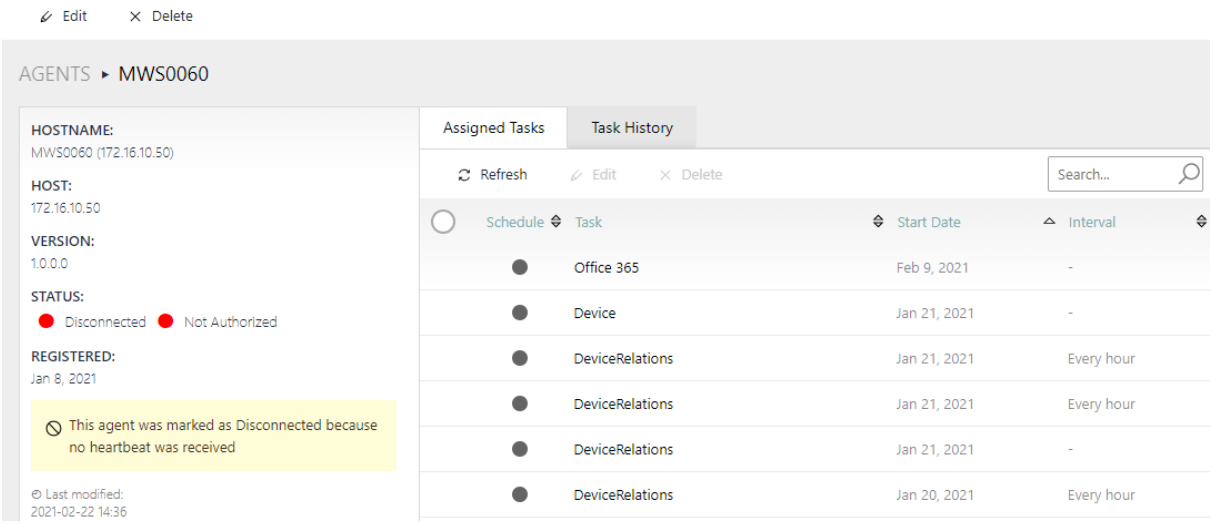


There is no way to add entries manually to this view. Instead, install a new agent using the provided MSI file. Once an agent is installed and started for the first time it automatically registers itself and will be visible in this view.



Agent Details

Clicking on an agent name in the Agents view navigates to the agent details. In this view the agent details and a list of assigned tasks to this agent is shown initially. The tab content can be changed by selecting a tab from the tab selection box. The history is shown when selecting the History-tab. The History shows all past and currently active tasks executed by this agent.



A list of recent changes made on the details of this agent are shown after pressing the link **Last modified** in the left part of the screen.

The agent can be edited and deleted using the action buttons located in the upper horizontal action bar:

- **Edit**
  Opens the "Edit Agent" panel that allows it to edit the current agent details

- **Delete**
  Deletes the current agent.

Note:
Deleting an agent is only possible when no task is assigned to it. A deleted agent can re-register at the RayVentory Data Hub server at any time if the agent service is still running. It must be explicitly reauthorized before it can request assigned tasks again

Editing Agents

Clicking on the **Edit** button from the actions bar in the agents (details) view opens a right side panel showing a form to edit the selected agent.

The following fields can be edited:

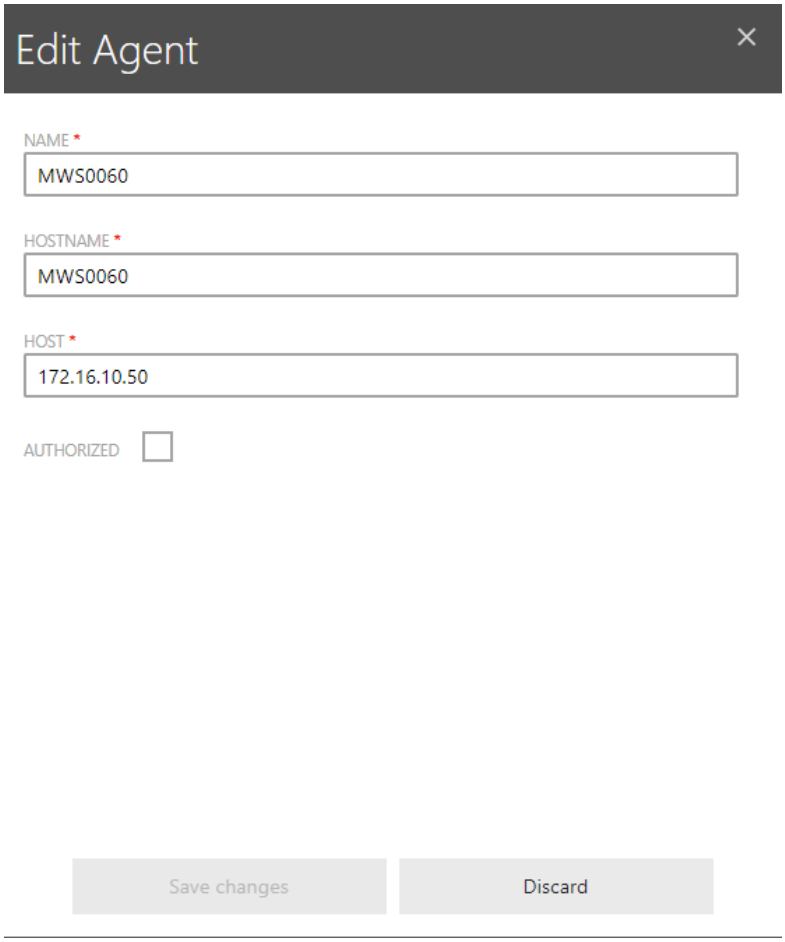
- **Name**
  The name of the agent.

- **Hostname**
  The hostname of the agent machine.

- **Host**
  The host address of the agent machine.

- **Authorize**
  Checkbox to authorize or revoke the agent.

Confirm the inputs by clicking on the button **Save changes** at the bottom. If the agent is successfully updated, a green pop-up notification is displayed in the top right corner.



If any error occurs, e.g. not all mandatory fields were filled or a field requires a unique name, a red toast notification is shown with further information. Furthermore, any invalid input field is marked with a red border and displays a short error message below the input field.

Tenant Settings

This screen is used to manage the settings which are tenant-specific.

General Tenant Settings

- **Agent expiration tolerance**
  The agent expiration tolerance is the maximum amount of minutes a task may start later than planned. If exceeded, the task expires.

- **Agent timeout tolerance**
  The agent timeout tolerance is an additional tolerance period to the task timeout period. When the task timeout and the timeout tolerance has passed the task is timed out.

- **Receive SQL Command timeout**
  The amount of seconds the save process for retrieved data is allowed to take before timing out.

- **Log entry amount**
  The maximum of log entries which should be kept for each logged entity.

- **Backup**
  The **BACKUP CURRENT DASHBOARDS AND REPORTS** button can be used to move all edited dashboards, reports, and tasks to a special backup folder. They will still be available in the library.

SMTP Settings (for Email Notification)

- **Server address**
  The address of the e-mail server.

- **Server port**
  The port used to communicate with the Email server.

- **Sender address**
  The address that will be put into the "From" field.

- **User account**
  The user name used to authenticate.

- **User password**
  The password used to authenticate.

- **Use SSL**
  Specifies whether to use TLS/SSL connection.

Email Reporting

Email reporting allows you to configure automatic notifications, containing selected reports. The notifications can be sent by e-mail to specific group of receivers. The receivers have to be users of RayVentory Data Hub (their accounts and e-mail addresses must be configured properly in the settings).

Prerequisites

- An accessible SMTP server (address and port)
- Credentials required for authorization with SMTP server

Before setting up the notifications, visit the **Tenant Settings** page, where SMTP server needs to be configured first.

Getting Started

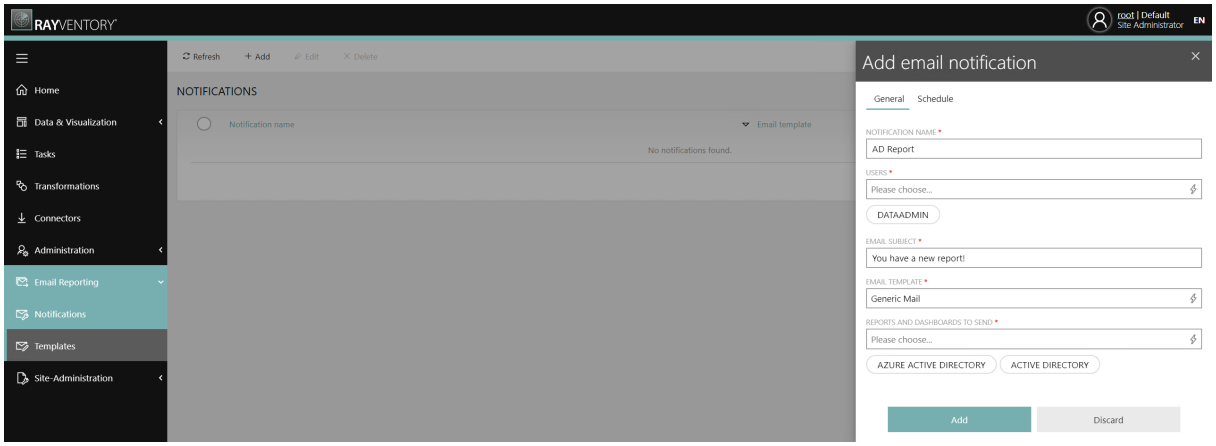
1. Ensure that the SMTP server is configured.
2. Set-up one or more templates, which will be used by the sending agent.
3. Add one or more notifications to link reports, templates and scheduling.

Notifications

Notification is a message sent to a user or a group of users, consisting of links to reports and/or dashboard in a user-specified template.

To add a new notification:

1. Ensure the prerequisites are met (see Email Reporting for more details).
2. Ensure there exists at least one template (see Templates for more details).
3. Press **Add** to open the notification editor:



Each notification has the following properties:

- **Notification name**
  Human-friendly name used for identification purpose. The receivers do not see this value.

- **Users**
  One or more groups of users to receive the message. The groups can be configured on the **Groups** page, and you can add users on the **Users** page.

- **Email subject**
  The subject of the e-mail.

- **Email template**
  A predefined template that will be used for the content. The templates can be managed and edited on the **Templates** page.

- **Reports to send**
  One or more reports / dashboards to send in this notification.

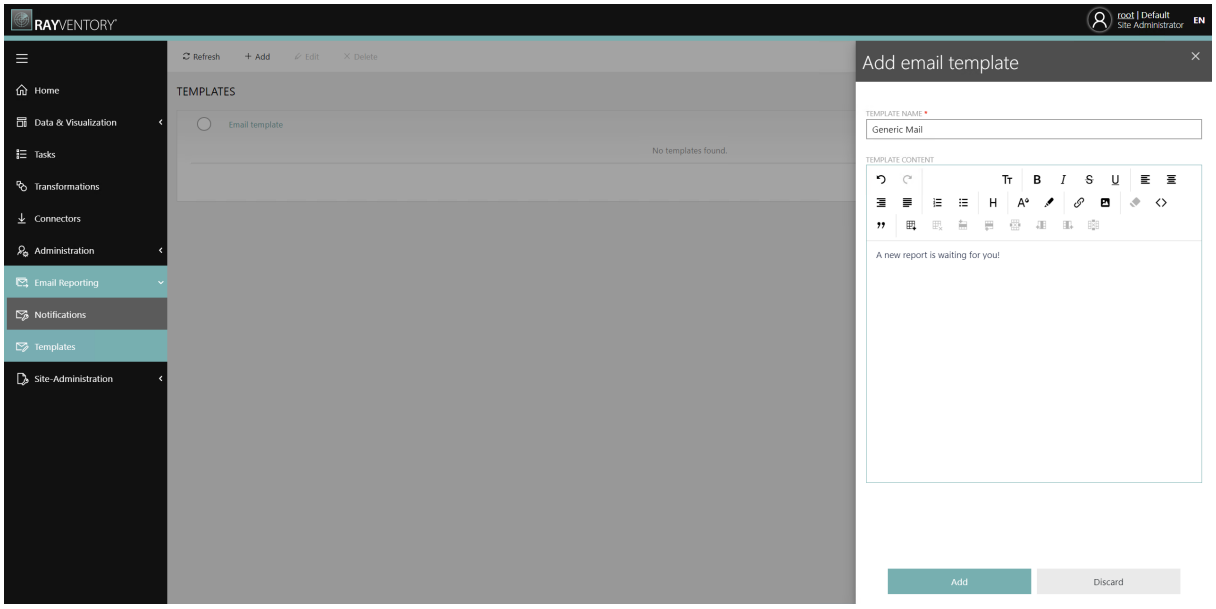
- **Schedule / Email notification active**
  Use this checkbox to activate the automatic notification.

- **Schedule / Interval options**
  Depending on the selected interval (previous drop-down) different options may be shown. If the **Advanced** interval is selected, a CRON expression is required as an input. Refer to internet sources, for example https://help.ubuntu.com/community/CronHowto to learn more about available options.

Templates

This is the place where templates for e-mail notifications can be configured.

A template defines the look and feel of the mail messages. You can have as many templates as required, and select the correct one on the **Notifications** page.



A visual, rich-edit WYSIWYG editor is provided.

Site Administration

This area is restricted to **site administrators** only.

It allows you to perform the following administrative tasks:

- Managing all users, in all tenants.
- Granting permissions, changing user roles.
- Managing tenants.
- Downloading and viewing system logs.

All Users

This view shows all users, defined globally for the current instance. It follows the same principles as the respective view Users, with the following additions:

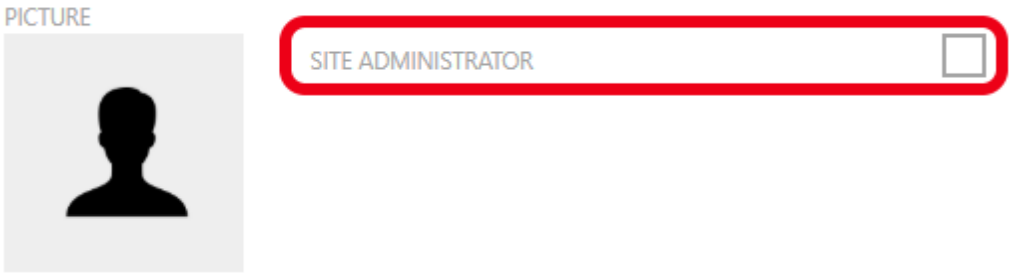
- It is not possible to define groups. Group configuration belongs solely to the project/tenant configuration.
- It is possible to promote any user to the **Site administrator** role. Such user will get extra permissions, including but not limited to defining new tenants, granting site permissions to other users or viewing system logs.

Other functions are similar to the usual tenant configuration page, section Users.

Defining Site Administrators

In order to promote a user to the Site Administrator role...

1. Ensure the user already exists. If the user does not exist, create a new user by pressing the + Add button.
2. Click on the required users to open the user view.
3. Both user view and the list show the current assignment. For a non-site-admin, the value should be No (cross).
4. Press Edit
5. In the new dialog, ensure the following checkbox is selected:



6. Confirm the change by pressing **Save changes**.

Tenants

Tenants are units, which provide a true separation of the reports, dashboards, users, groups and other relevant-settings. Since - unless an explicit access to another tenant is granted - every user sees only his data, it is an effective way of having a single instances installed once, which serves different customers, projects or any other entities.

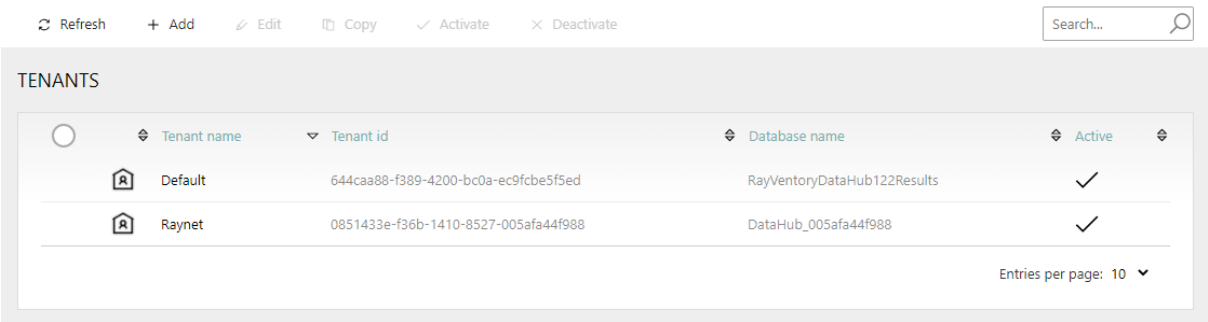
Every RayVentory Data Hub has at least one tenant - the default one. It is the only tenant that is created automatically and requires no further actions. Tenants are not only, but also physically separated.

- **Logical separation**
  The data, reporting objects, users, groups and other settings are tenant-specific, and only shown to the users who have access to respective tenants.

- **Physical separation**
  The reporting data is physically separated, by using a different database. This way, even if a database for one tenant is compromised, the other tenants and their databases (which potentially may contain sensitive data) are not affected. Every tenant has a unique encryption for his data (optional feature), which means that the attacker having access to the data sees only scrambled values, and without a proper key is not able to decipher it.

Managing Tenants

The **Tenants** view provides a convenient way of viewing, adding and editing tenants.



Each tenant is described using the following properties:

- **Tenant name**
  This is a value used for displaying purposes. Tenant name is used as a caption in tenant selector (see Login and The Header).

- **Database name**
  This is the name of the database where the reporting data will be stored. Each tenant should have an unique database, to ensure the data is separated and only users belonging to the right tenant can access it. The database name comes from the underlying connection string. The connection string defines the actual server, instance, database name and other relevant parameters.

- **Active**
  Tenants can be active or disabled. Disabled tenants are not shown in tenant selector (see Login and The Header). Site administrators can see all tenants, including disabled ones.

Other parameters (not visible directly in the grid):

- **Connection string**
  The full connection string, providing access to the database with tenant's data. This has an impact on the database name column.

- **Encrypt data**
  This setting defines, whether the reporting data is transparently encrypted. This feature uses Microsoft SQL Server **Always Encrypted** functionality. It makes sure that the data in the SQL tables is scrambled and obfuscated (on cell basis), but the users who have access to the correct tenant see the actual data, both in reports and during the designing. See chapter Data encryption for more information about it.

Data Encryption

RayVentory Data Hub supports data encryption of the following information:

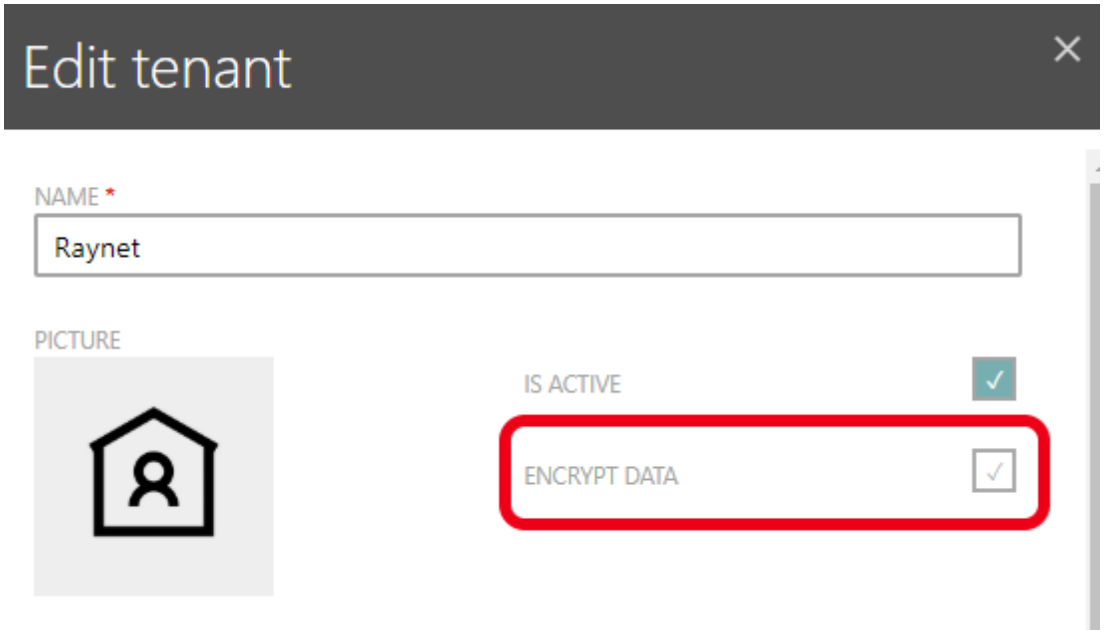
- Sensitive data belonging to tasks, including:
  - connection strings,
  - password,
  - logins,

  - client secrets,
  - tokens
  - etc.
- Reporting data

The encryption of the former is using tenant-specific keys. The encryption of the latter relies on Microsoft SQL Server **Always Encrypted** functionality. It makes sure that the data in the SQL tables is scrambled and obfuscated (on cell basis), but the users who have access to the correct tenant see the actual data, both in reports and during the designing. Always Encrypted is a feature designed to protect sensitive data stored in SQL Server databases. It provides an extra separation level between the owner of the data (capable of writing and reading to it) and the maintainer (who should not be able to decipher the content). The encryption key are managed by RayVentory Data Hub and never revealed to the SQL Server. This means, that in case of a compromised database with sensitive data, the attacked is unable to decrypt it.

Note:
Data encryption is only supported by SQL Server Enterprise. It may not be supported on all editions and versions of SQL Server. Make sure that the underlying database engine supports it, before activating the function.

Data encryption is not enabled by default. In order to enable it, locate the tenant to be secured, and press **Edit** to open the edit dialog. The dialog has several options, the one relevant for data encryption is the following checkbox:

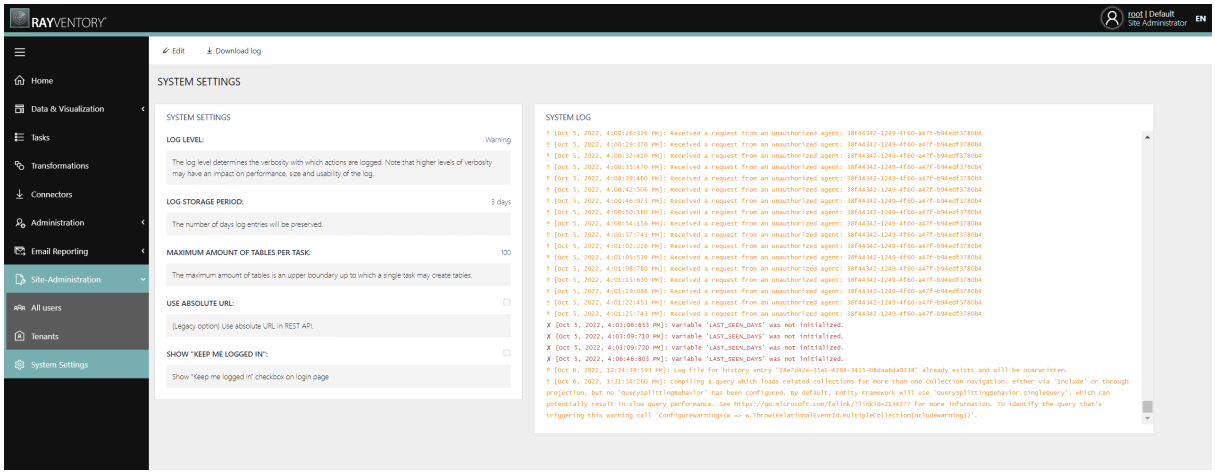


Once the setting is active, the set-up is done and the data is protected.

Note:
Data encryption adds extra security, but not without costs. It is expected that querying, filtering, sorting and other SQL operations may get a performance hit, due to the fact that every affected cell, column and row must be decrypted before displaying, sometimes requiring extra steps or round-trips. The impact on the performance depends on the amount of the data, the level of the optimization of particular reports, and on the hardware on which the host is running.

System Settings

This is the place where system settings of RayVentory Data Hub can be changed. The screen is accessible by only Site Administrators.



The **Download log** button can be used to download the log file to the local machine. The log file can also be found on the server in the following location:

- `[RVDH-INSTALL DIR]\logs\RayVentoryDataHubService.log`

At the bottom of the page the latest log entries from the log of the backend can be found at the bottom of the page in the **SYSTEM LOG** field.

The **Edit** button on top of the page can be used to open the **Edit Settings** dialog.

In the **Edit Settings** dialog the following options are available:

- **LOG LEVEL**: This field can be used to choose the log level using the dropdown menu. The log level determines the verbosity with which the actions are logged. The following log level are available in the dropdown menu.
  - All
  - Debug

- Fatal
- Info
- Off
- Warning

Note:
Higher levels of verbosity may have an impact on performance, size, and usability of the log.

- **LOG STORAGE PERIOD**: This field defines the period that the log will be stored. The log entries will be preserved and kept ready for download for the period of day(s) that is defined in the field.
- **MAXIMUM AMOUNT OF TABLES PER TASK**: This field defines the maximum amount of tables that can be created by a single task.
- **USE ABSOLUTE URL**: If the checkbox is checked, the absolute URL will be used in the REST API. This is a legacy option.
- **LDAP DOMAIN NAME**: If LDAP is used, the domain name that LDAP will use needs to be entered here. More about LDAP can be found here.
- **SHOW "KEEP ME LOGGED IN"**: If checked, the **Keep me logged in** checkbox on the login page will be shown. Otherwise, the checkbox will not be shown.

Guides and How-Tos

This chapter contains detailed step-by-step guides and recipes for some more complex tasks.

Connecting to OracleDB with ODBC Connector

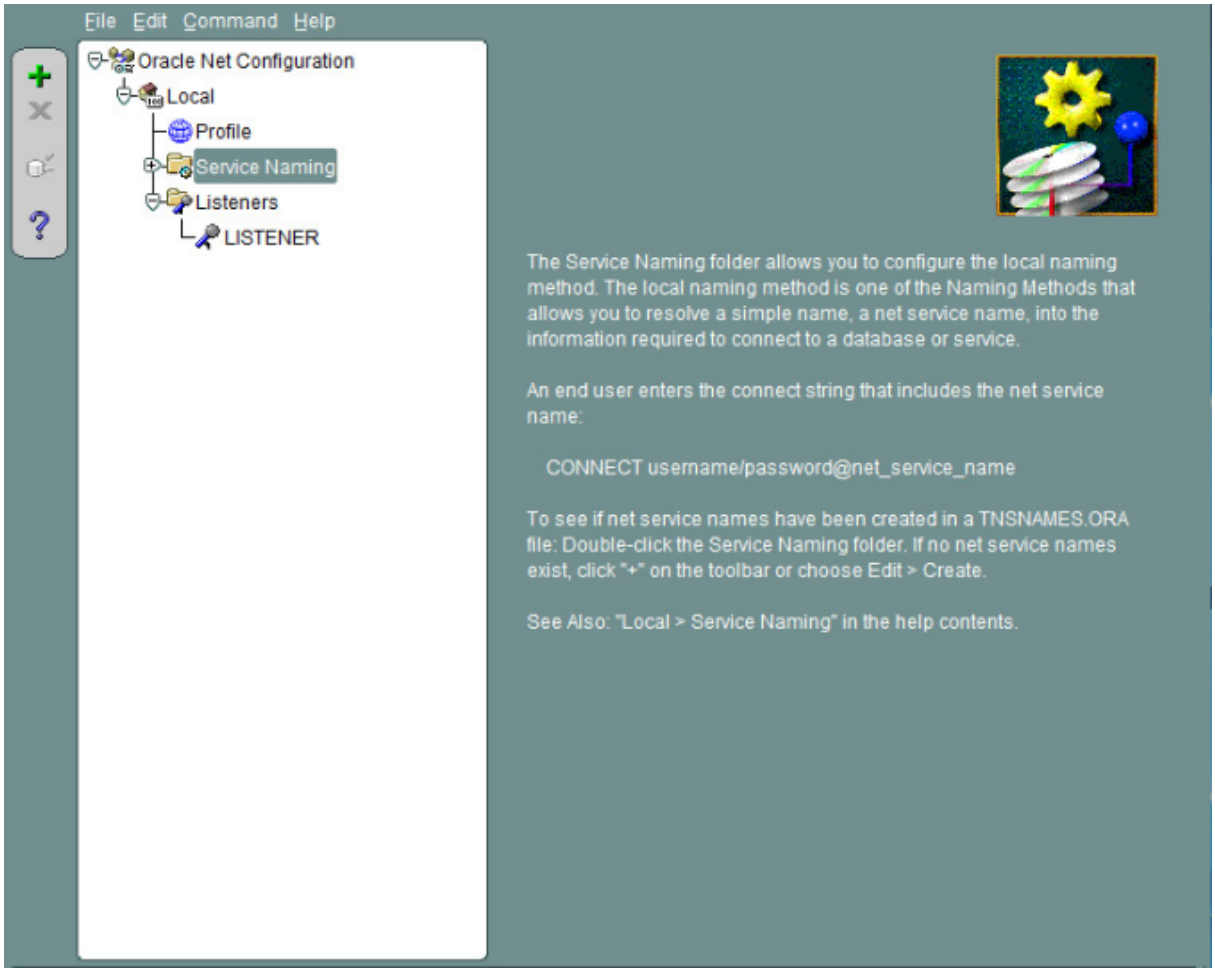
Introduction

This step-by-step guide shows how to configure RayVentory Data Hub to query information from Oracle database with DSN using an ODBC Oracle Instance Client Driver.

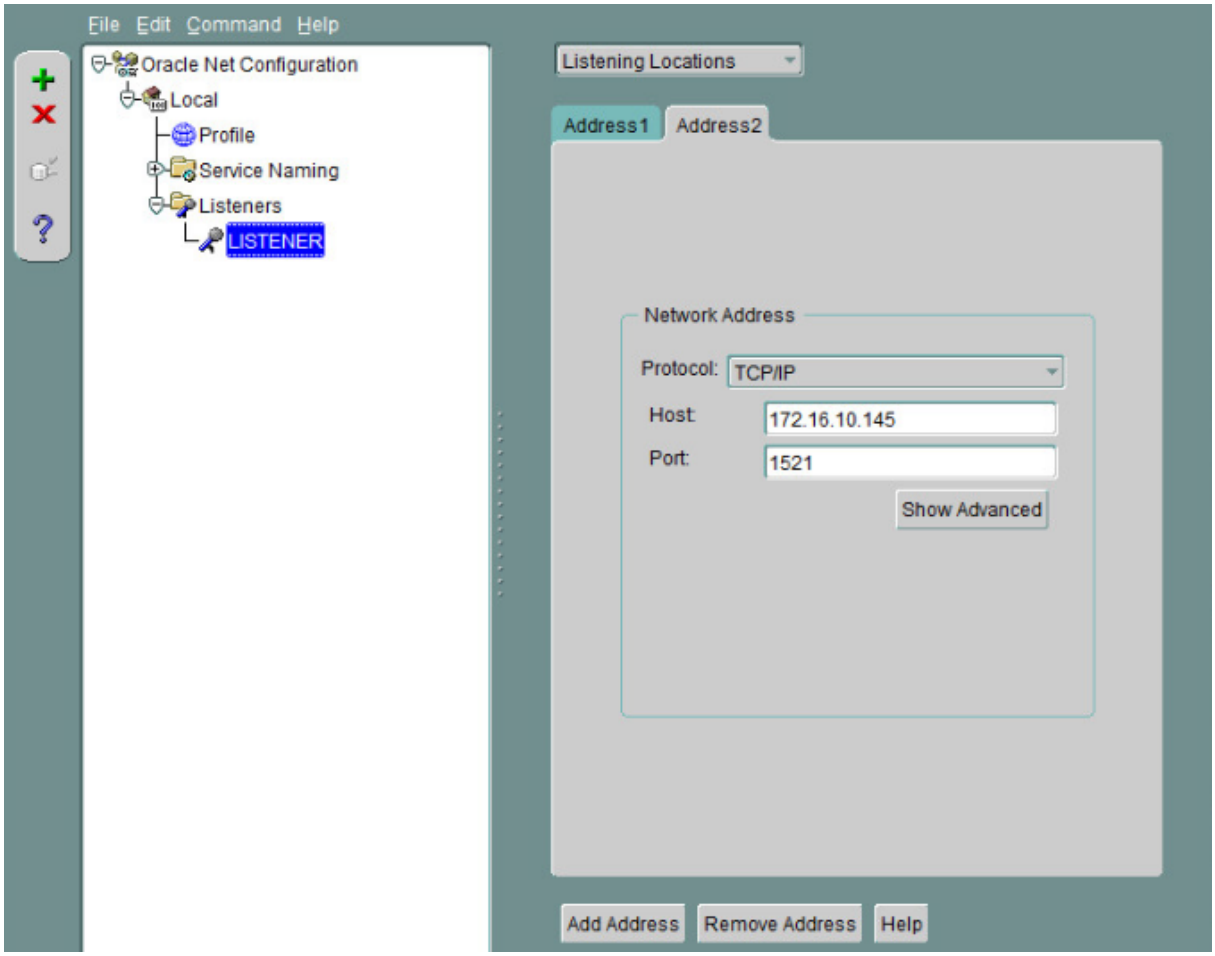
Prerequisites

On the Target Machine

- A database user with read permission on the Oracle database is available.
- Add a Service Naming and a Listener using the Oracle Net Manager.
  - Service Naming:
    - Add a Service Naming and take note of the Service Name, Protocol, Host-Name and Port. This information will be required later.



- Listener:
  - Add a Listener listening on the Host-IP where the Oracle Database resides.



Configuring the Client Machine

1. Download the basic-package instantclient-basic-windows.x64-19.6.0.0.0dbru.zip
   Download page: https://www.oracle.com/de/database/technologies/instant-client/winx64-64-downloads.html

2. Extract the zip file (it should contain a folder called `instantclient_x_y`).

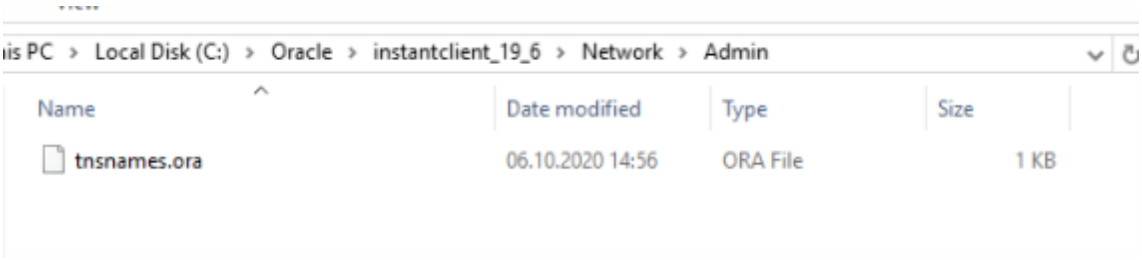
3. Download the Instant Client ODBC package instantclient-odbc-windows.x64-19.6.0.0.0dbru.zip

4. Unpack it in the same directory (`instantclient_x_y`) as the Basic package.

5. Run `odbc_install.exe` from the Instant Client directory.

6. Reboot your machine when prompted.

7. Go into the `instanclient_x_y` folder and create a subfolder `Network\Admin`



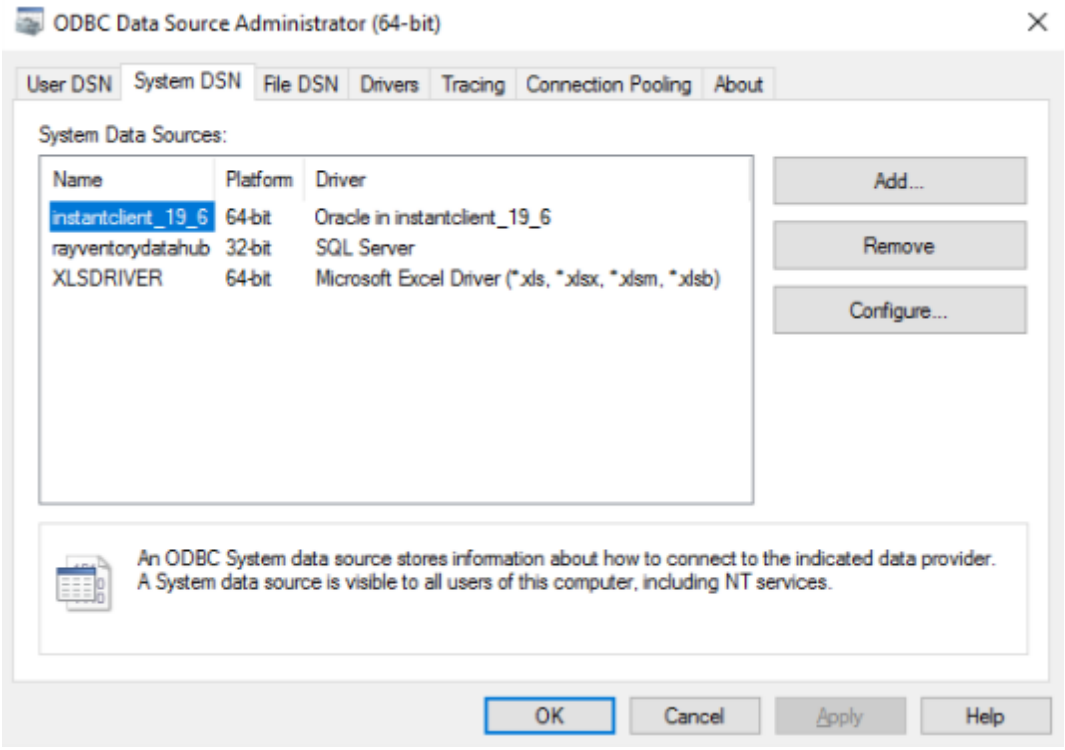
8. Write the following content and save it as `tnsnames.ora` in the above folder:

```
[TNSALIAS] =
       (DESCRIPTION=
           (ADDRESS_LIST=
               (ADDRESS=(PROTOCOL=TCP)(HOST=[HOST])(PORT=[PORT]))
            )
           (CONNECT_DATA=
               (SERVICE_NAME=[SERVICENAME])
            )
        )
```

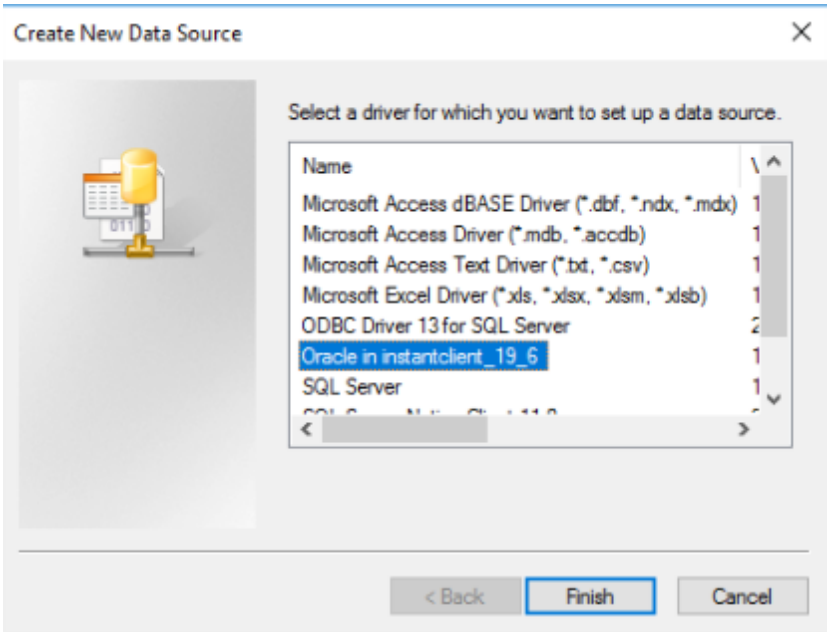
9. Replace...

   - [TNSALIAS] with anything you like. It will be needed during the configuration of the ODBC Driver.
   - [HOST] with the host of your target machine.
   - [PORT] with the port of your target machine.
   - [SERVICENAME] with the service name configured on your target machine using Oracle Net Manager.

10. Open the *ODBC Data Source Administrator (64-bit)* and select the tab System DSN



11. Click on **Add...** and select `Oracle` in `instantclient_x_y`. Click **Finish**.

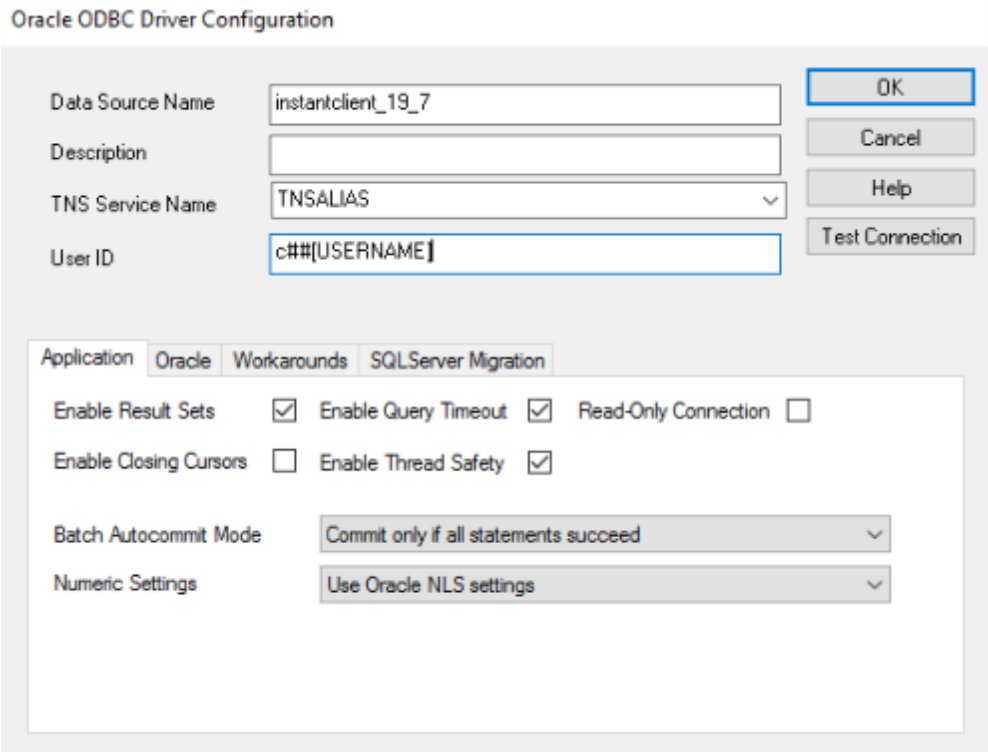


12. A new window opens called Oracle *ODBC Driver Configuration.*

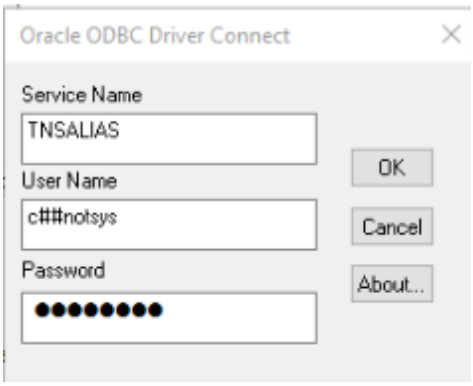
- In the **Data Source Name** field type any name you like. You'll need it later in RayVentory Data Hub when specifying the connection string for a task.
- In the **TNS Service Name** field provide the same TNS alias name you provided in step 9.
- In the **User ID** field type in `c##[USERNAME]`, where `[USERNAME]` must be replaced by a

username that has read access on the Oracle Database.

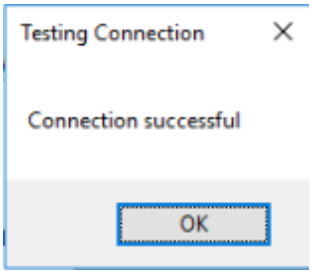
- Leave any other setting untouched.



13. Click **Test Connection**. Provide the password of the Oracle database user.



14. If the connection was successful the following window pops up:



15. Click **OK**.

16. Remember the value of the **Data Source Name** and click on **OK** in the **Oracle ODBC Driver Configuration** Window.

17. Install and configure Data Hub Agent, as described in the following chapter: Installation and configuration.

Configuring Data Hub Settings

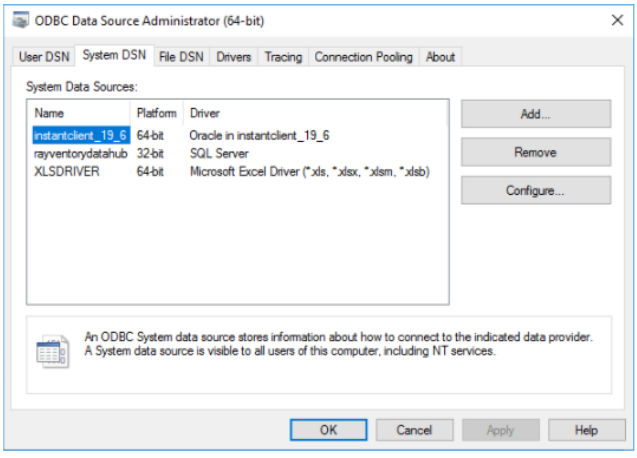
1. Sign in to RayVentory Data Hub.

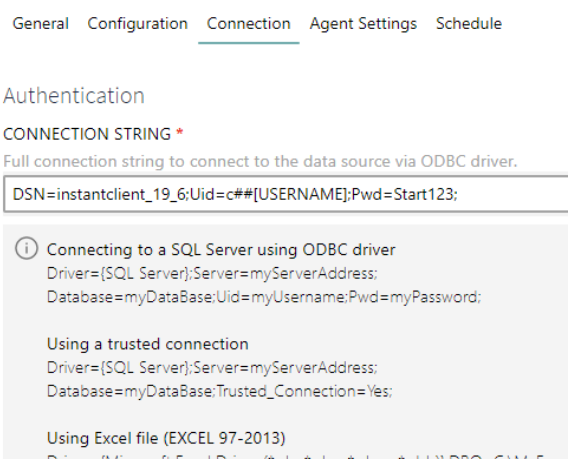
2. Create a new task using ODBC as the connector type.

3. Fill in the required fields in the *General*, *Configuration*, *Connection* and *Agent Settings* tabs. In the *Connection* tab use the following connection string, that connects to the oracle db through DSN:

   ```
   DSN=[Data Soure Name];Uid=c##[USERNAME];Pwd=******;
   ```

   where `[Data Source Name]` is replaced by the data source name you configured within the ODBC Date Source Administrator and `[USERNAME]` is replaced by the username of the user with read access on the Oracle database.



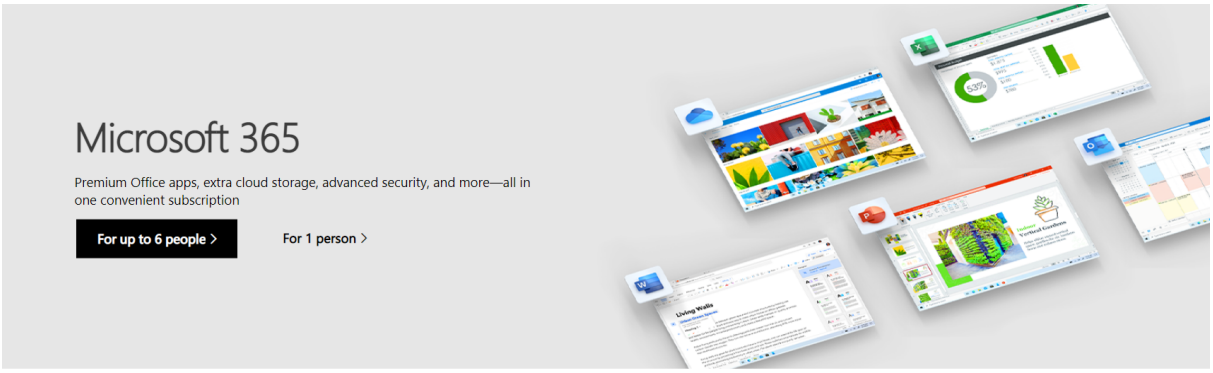


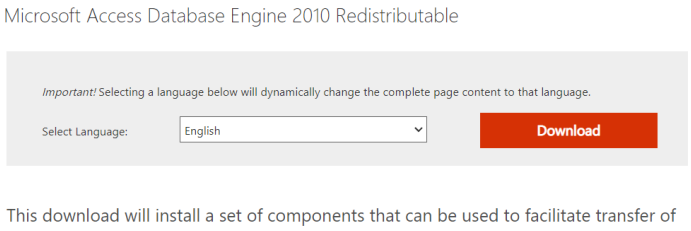
Setup the ODBC Connector for Excel

This step-by-step guide describes how to setup the ODBC connector in order to work with Microsoft Excel.

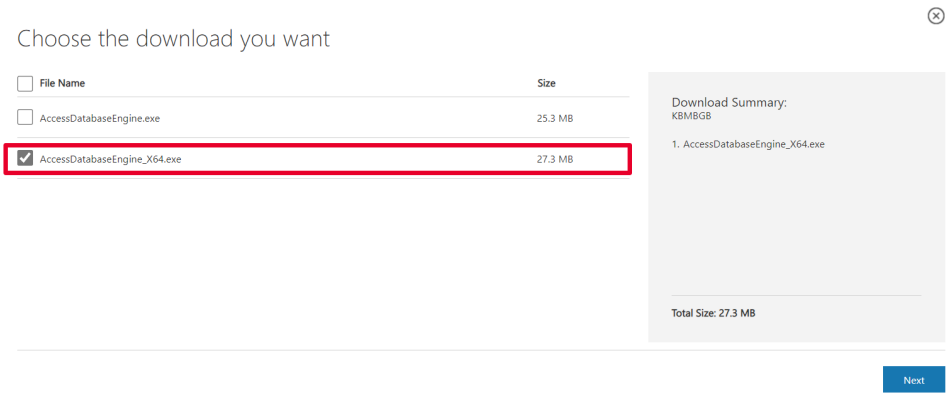
Prerequisites

In order to use the ODBC connector with Excel it is necessary to download the latest ODBC Office version. The latest version can be found under https://www.microsoft.com/en-US/download/details.aspx?id=13255.



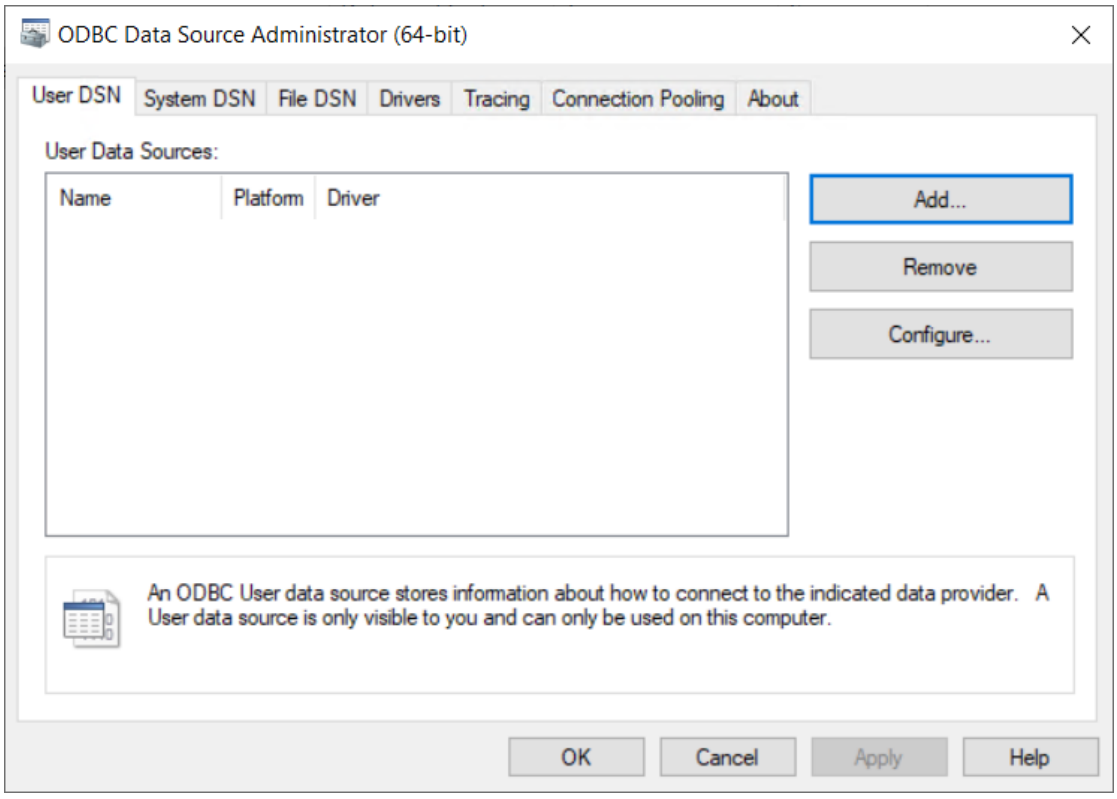


On the page, click on the **Download** button. The following menu will be shown:

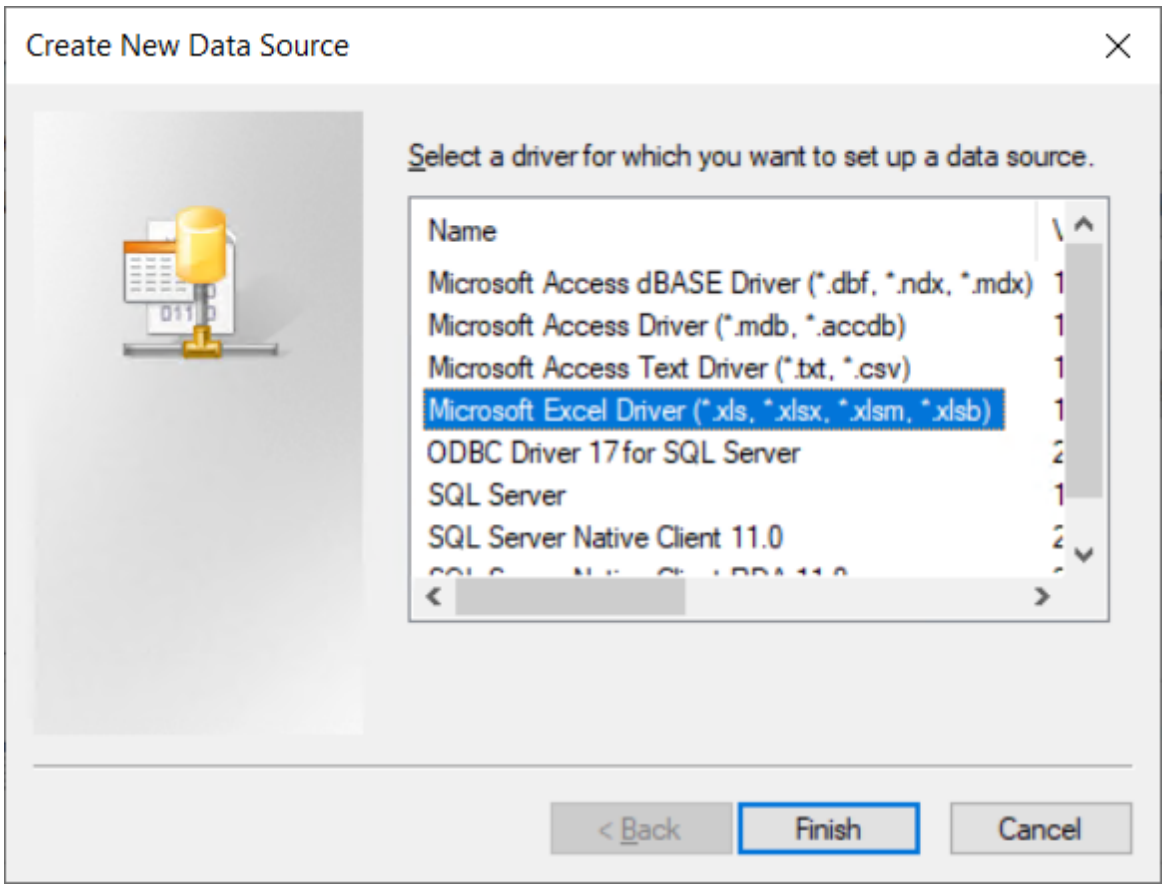


Select `AccessDatabaseEngine_X64.exe` and click on the **Next** button. After downloading the file, install the AccessDatabaseEngine in order to get the latest drivers.

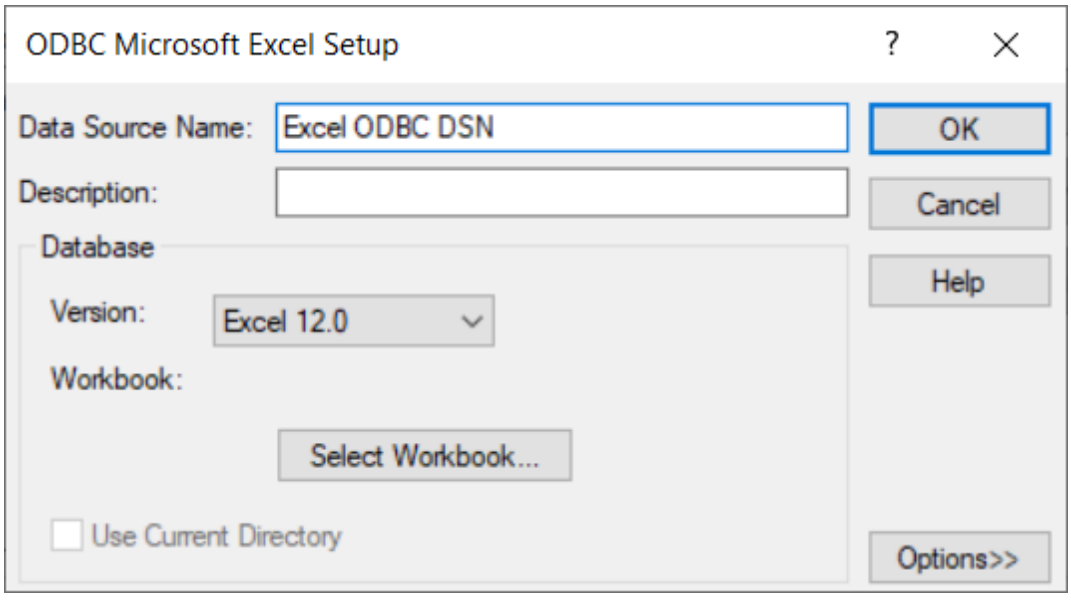
Open the Start menu and search for ODBC. Select the ODBC Data Sources (64-bit).



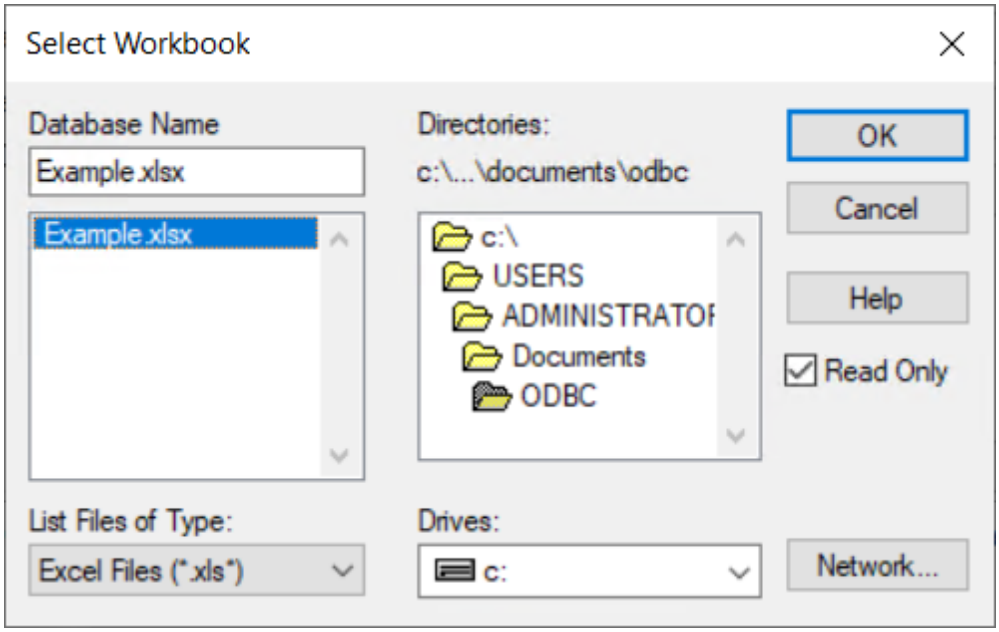
In order to create a new ODBC User data source for usage with the ODBC connector click on the **Add...** button while in the **User DSN** tab.



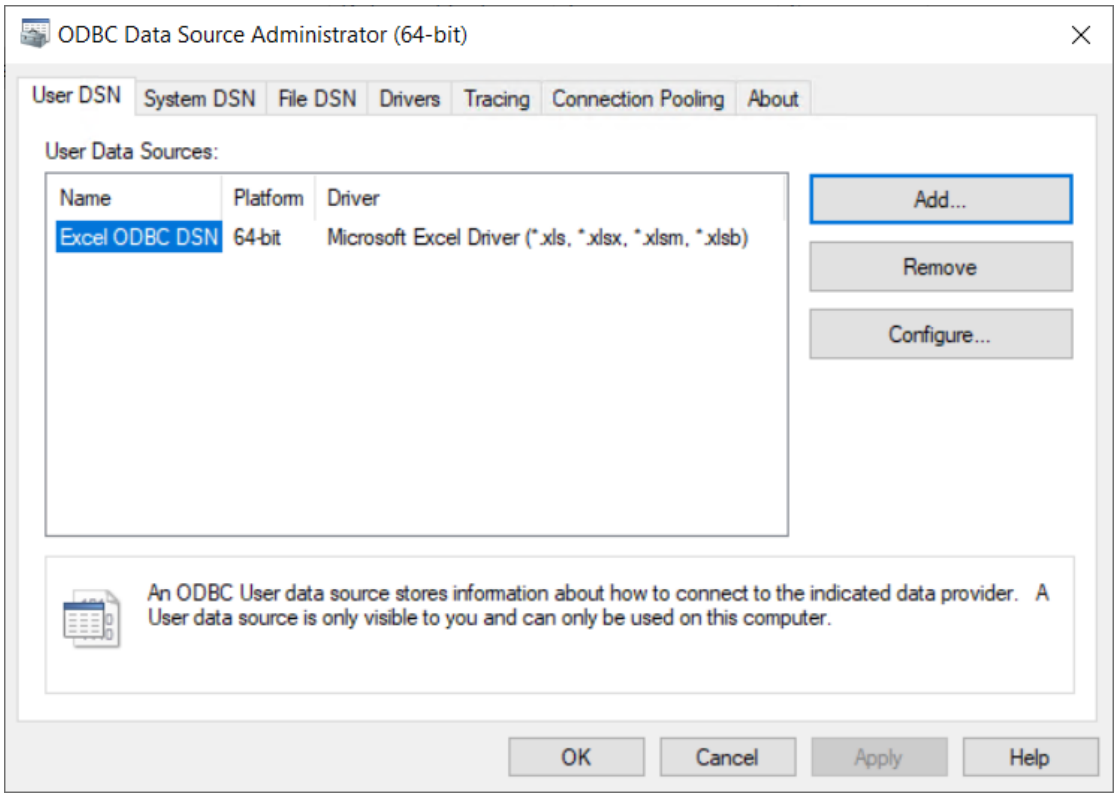
From the list of drivers, select the **Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)** and click on the **Finish** button.



Enter a Data Source Name and ensure that Excel 12.0 is selected. Click on **Select Workbook...** in order to select the target Excel file.



In the Select Workbook dialog, go to the directory where the target file is located and select the file from the field located below the **Database Name** field. Click on the **OK** button. After returning to the ODBC Microsoft Excel Setup dialog, click on the **OK** button.

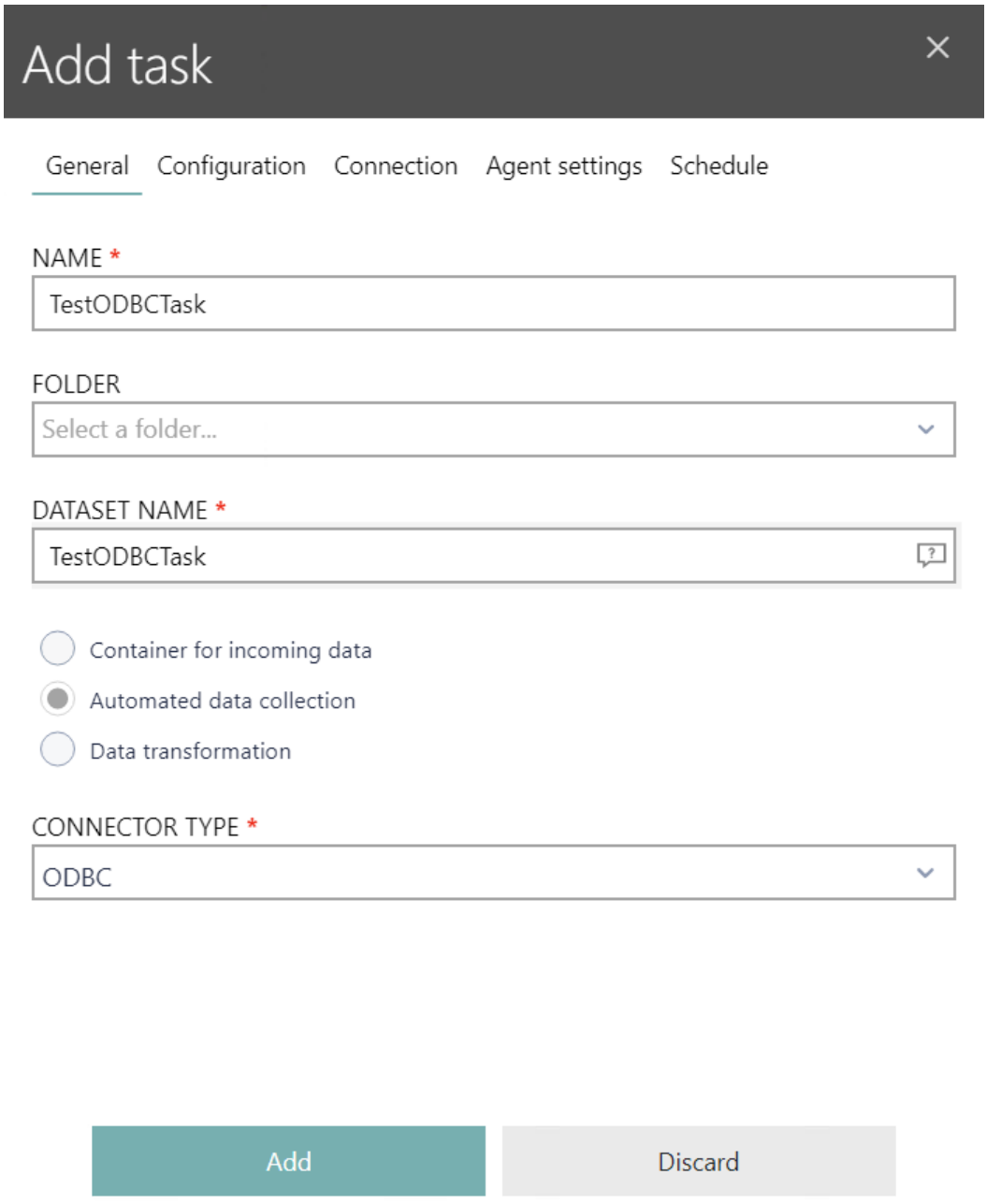


The newly created data source will now be available for selection. Select the data source and

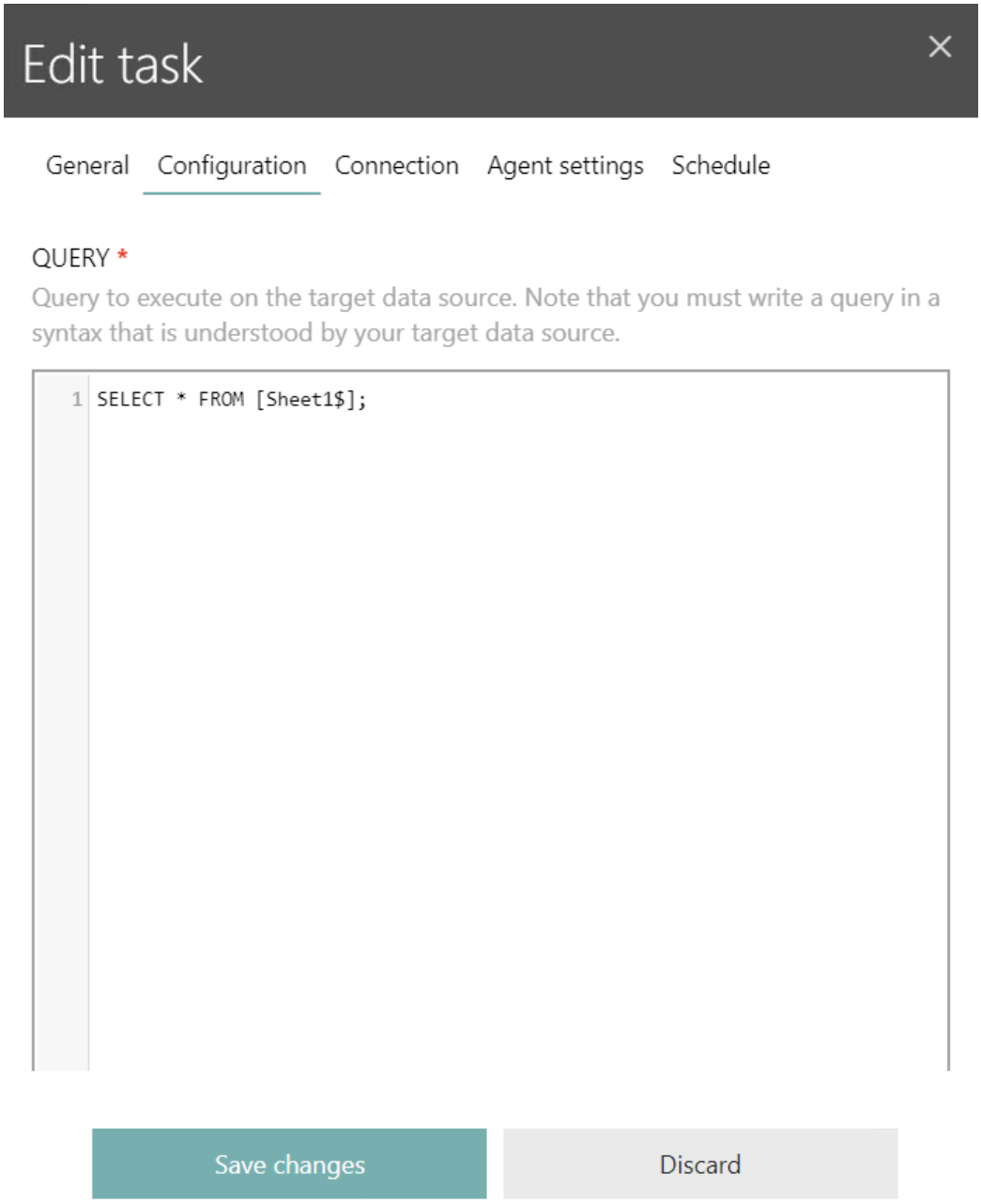
click on the **Add...** button to finish the setup.

Setup of the ODBC Connector in Data Hub

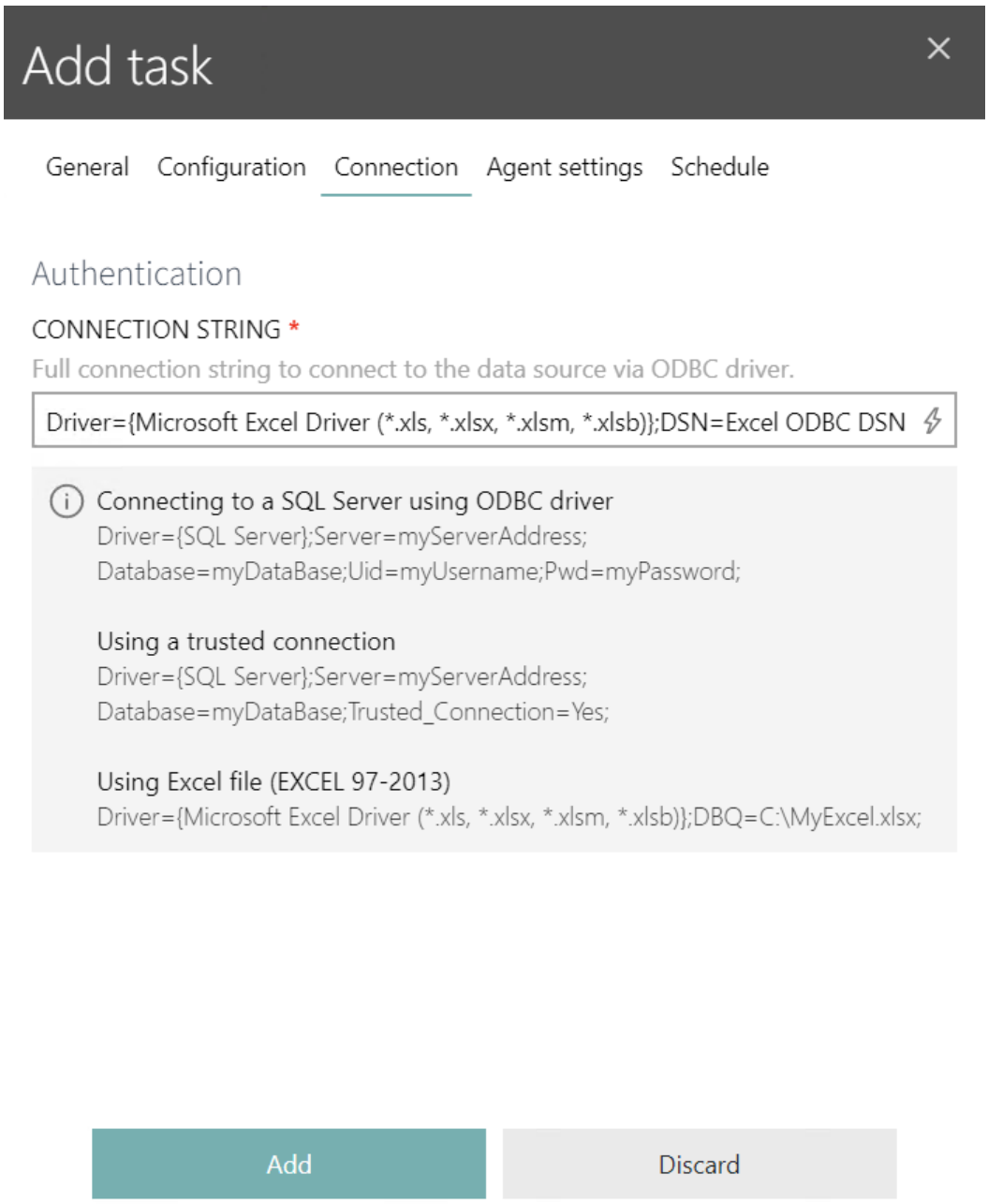
In the Tasks section of Data Hub, click on the **+ Add** button. The **Add task** dialog will be opened.



In the **General** tab, enter a name for the Task into the **NAME** field and a name for the dataset into the **DATASET NAME** field. Select ODBC as **CONNECTOR** type and switch to the **Configuration** tab.



In the **Configuration** tab, enter the query that is to be executed on the data source. One sheet of an Excel file represents a table. In order to target a sheet, the following syntax needs to be used: `[NameOfTheSheet$]`. After entering the query, go to the **Connection** tab.



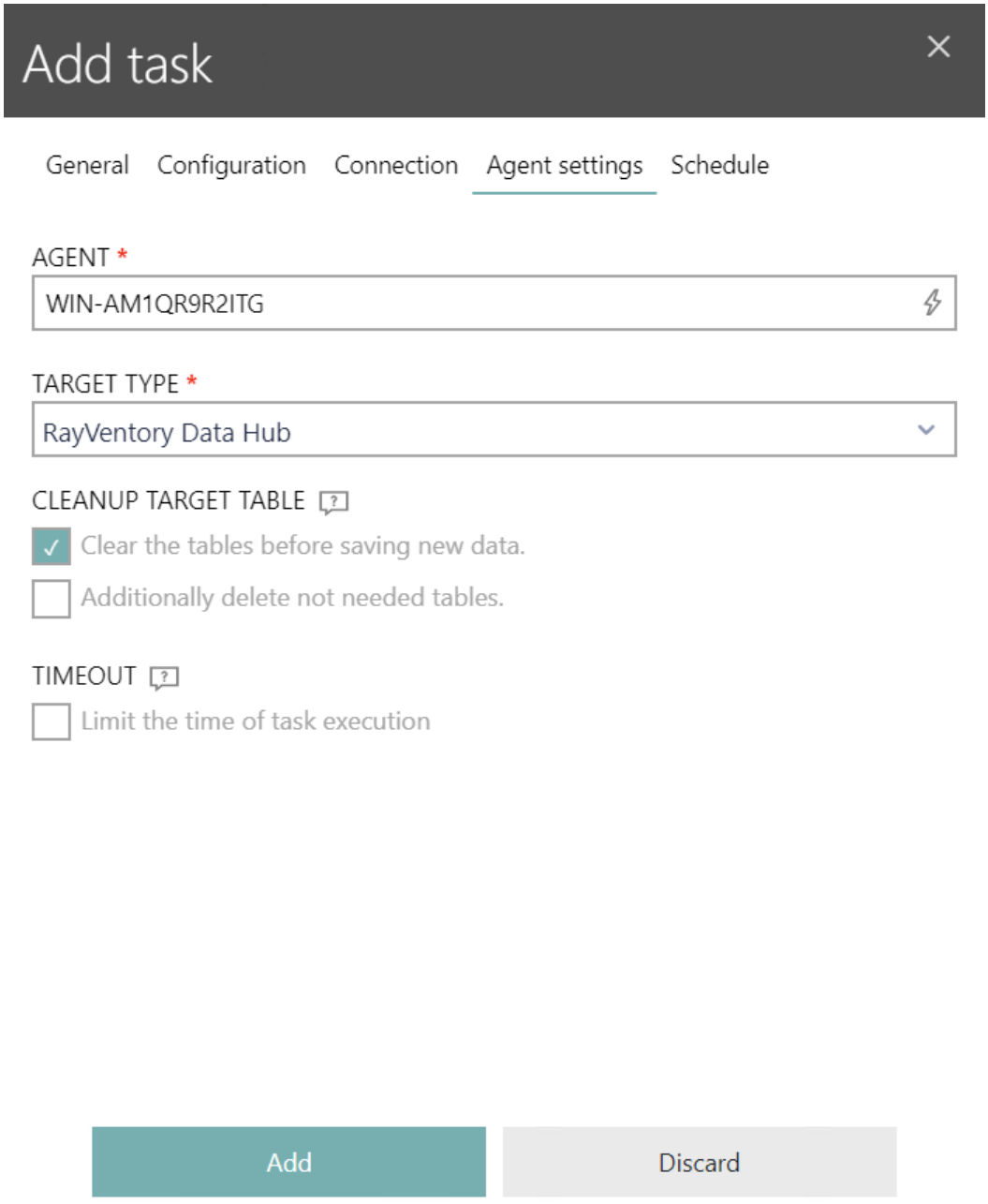
Enter the full connection string into the **CONNECTION STRING** field. There are two ways to create the ODBC connection string.

1. Set the DSN name to the name that was defined in the ODBC Driver view.
Example:
```
Driver={Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)};DSN=Excel
ODBC DSN
```

2. Use the path to the Excel file as connection string.
Example:
```
Driver={Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)};DBO=C:
```

```
\Users\Administrator\Documents\ODBC\Example.xlsx
```

After entering the connectin string, switch to the **Agent settings** tab.



Enter the agent that will be used into the **AGENT** tab and click on the **Add** button.

Using PowerShell Connector

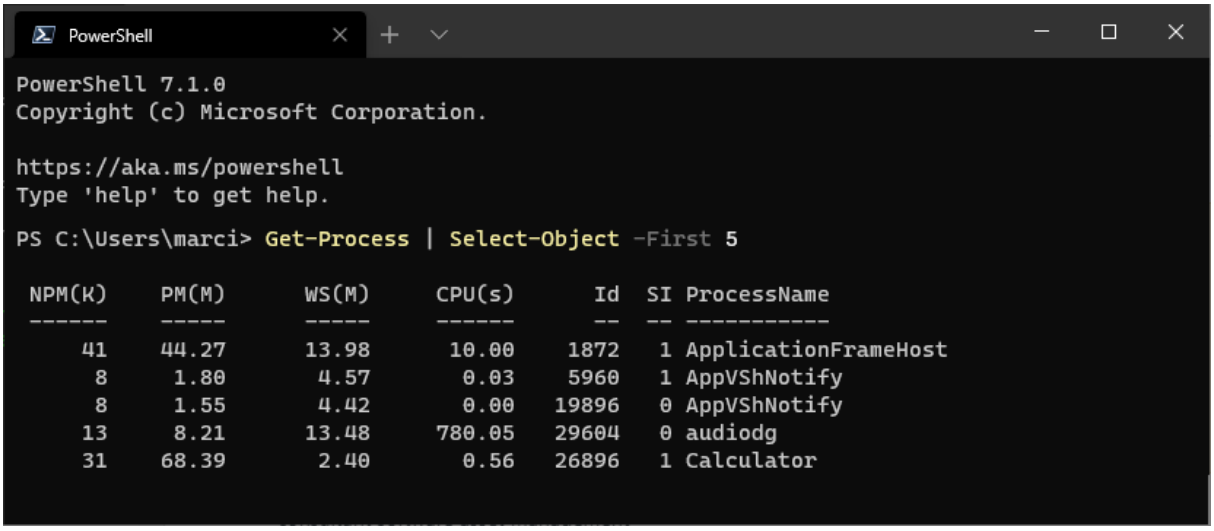
PowerShell connector enables configuring custom or complex tasks, which can be automated by using one or more PowerShell commands. Because PowerShell can be also used to start any

executable, this connector can also be used to query the data from various system tools and other CLI-based interfaces.

Understand the risk:
PowerShell code that you write is executed with the permissions of the service user that the agent is running. The code can import other modules and execute all commands from modules in scope. This may be a potential security considerations, because anyone having access to the administration of tasks can use this feature to remotely execute the code on the agent machine.

In order to create a task that fetches the data with PowerShell connector:

1. Write the content of your script and test it locally with PowerShell 7. For example, the following will print first 5 processes running on the machine:

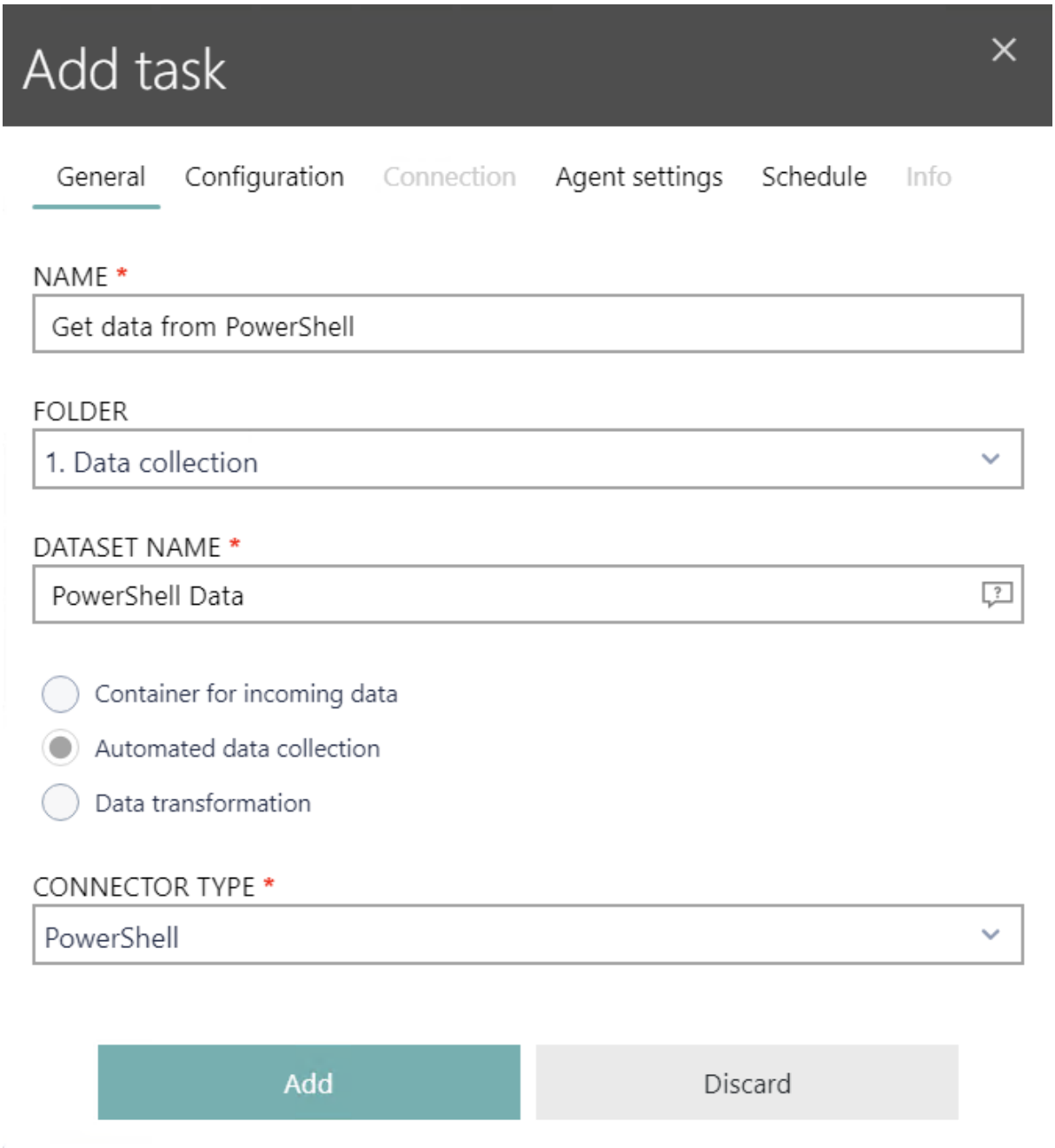


Note:
For testing, piping the results to command lets like `Format-Table` or `Format-List` is acceptable. However, when using from RayVentory Data Hub it is important that the objects are returned to the output (and not their formatted output). This is why in the final script that is executed by RayVentory Data Hub you should avoid using extra formatting.

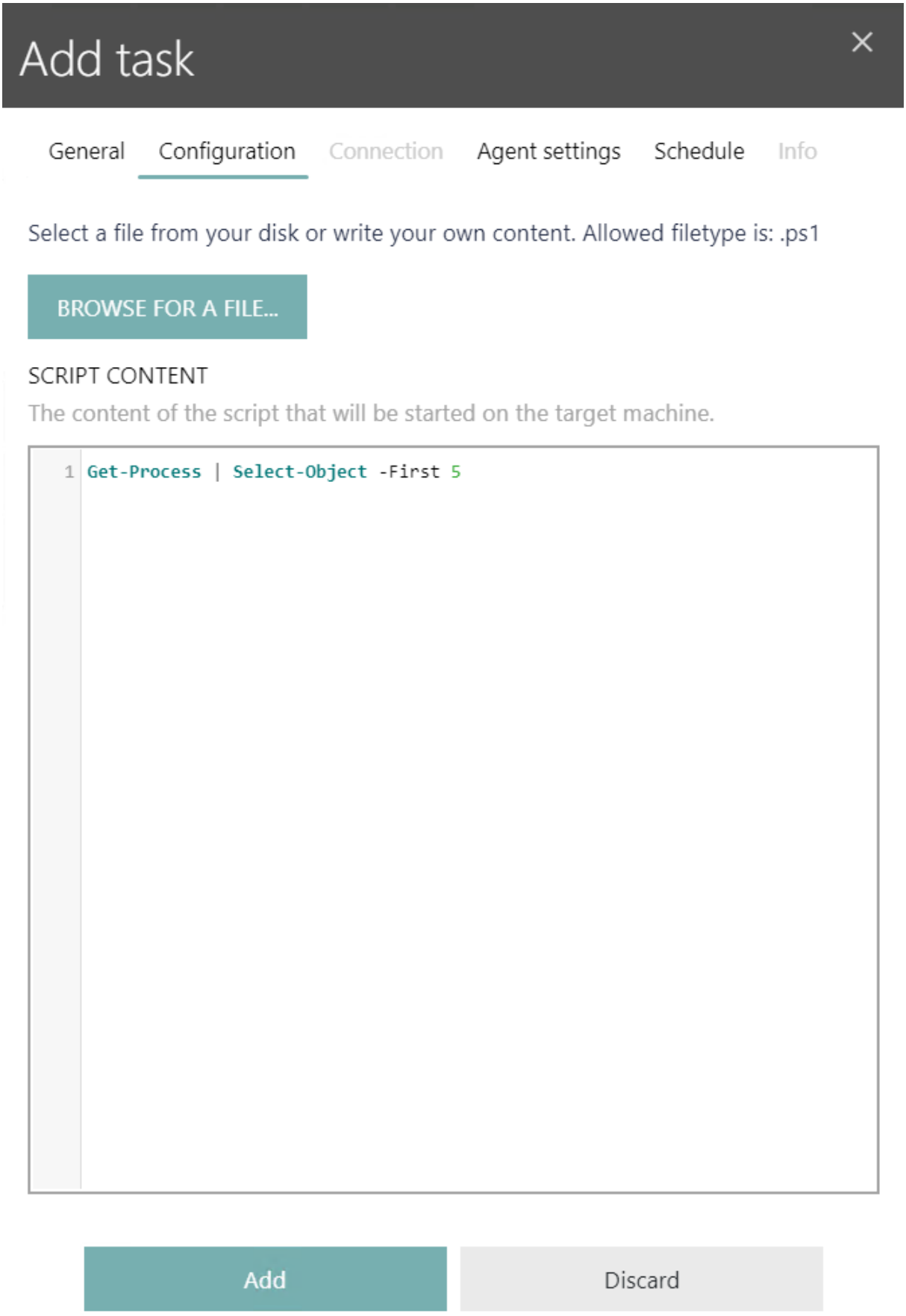
2. The script should simply output the desired results. Make sure that you do not include output that you do not want. For example, many command lets will print to the output if the results of their execution is not assigned to a variable. The simplest way to ensure that everything is well-prepared is to run the script and verify that there are no unnecessary objects, like in the example below.

3. If required, you can import external modules with `Import-Module`. Bear in mind, that the module must be available on the agent machine which will later run the script.

4. Once the complete script is ready (including all required import statements and writing the results to the output) you should create a new task of type PowerShell and put your script there:



5. In the **Configuration** tab, enter the content of your script.


Add task
General
Configuration
Connection
Agent settings
Schedule
Info
Select a file from your disk or write your own content. Allowed filetype is: .ps1
BROWSE FOR A FILE...
SCRIPT CONTENT
The content of the script that will be started on the target machine.
1 Get-Process | Select-Object -First 5
Add
Discard

Note:
RayVentory Data Hub has a limited syntax-highlighting support, but otherwise the debugging or formatting options are limited. You can use your favorite IDE (for example Visual Studio Code) to author and debug the script before you put its content in the product configuration.

6. Configure the rest of options as required, and finally start the task.

Working with Multiple Tables

By default, all returned objects are written to the same task. To work with multiple tables, use the custom command let Set-DataHubTableName which is available in your PowerShell session. You should use this command to inform the collector that a new table is about to begin, with a name being passed as the only argument. For example, the following gets processes and packages and writes them to two different data sets:

```
Set-DataHubTableName "Processes"
Get-Process | Select-Object -Property Name
Set-DataHubTableName "Packages"
Get-AppxPackage -Publisher *Microsoft*
Get-AppxPackage -Publisher *Raynet*
```

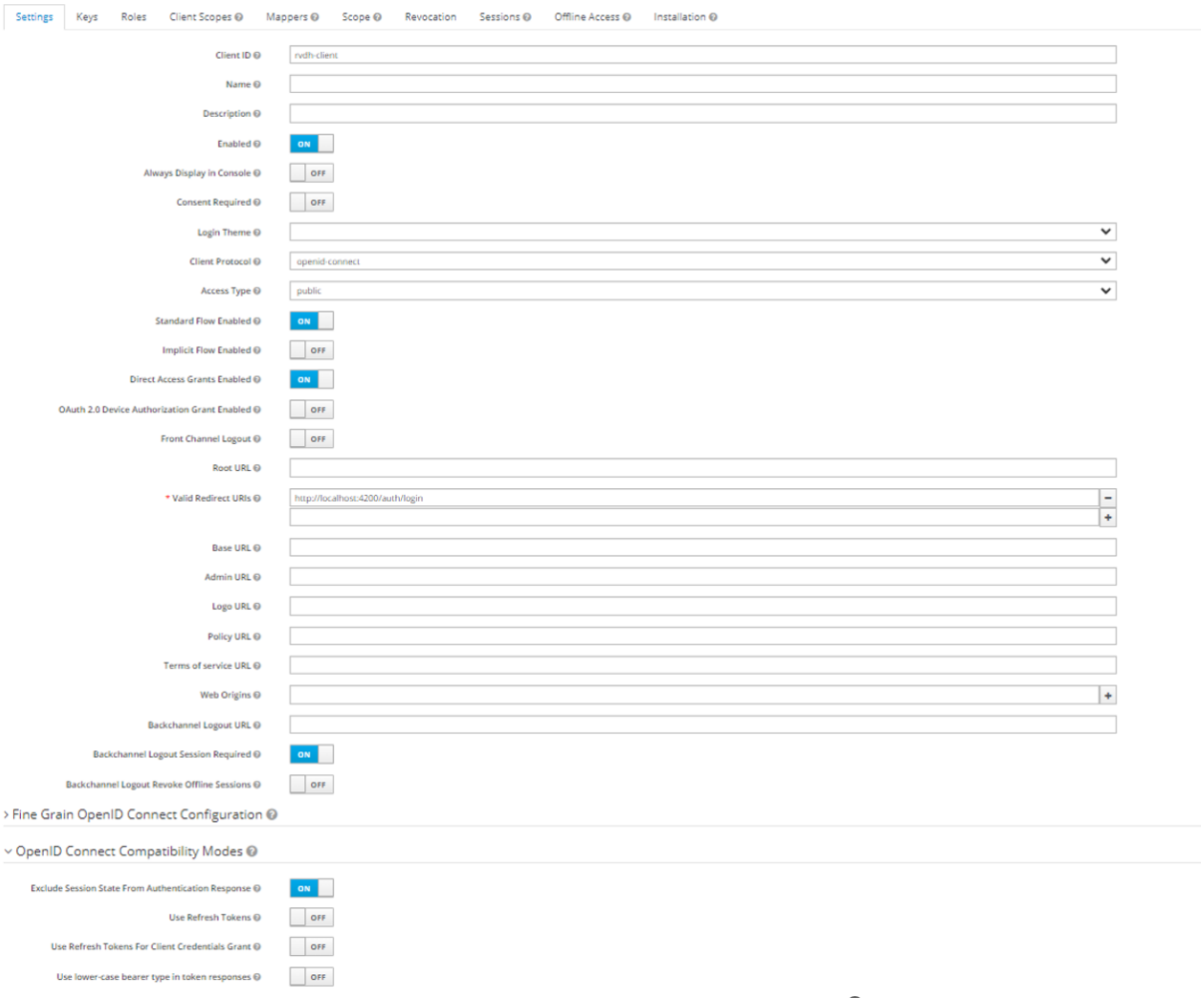
If your PowerShell task uses data set name PowerShell, then as a result the following will be collected by the PowerShell connector:

- Table `PowerShell.Processes` with the names of currently running processes.
- Table `PowerShell.Packages` with the list of packages from Microsoft and Raynet, with all properties returned by the command let.

This also demonstrates that subsequent calls are grouped and injected into the most recent table.

How-To Configure KeyCloak

In order to configure the KeyCloak login for usage with RayVentory Data Hub it is necessary to create a new Client in an existing or new KeyCloak realm.



Add the mandatory information to the settings page of the client and set the following settings:

- Main Settings
  - **Enabled** = "On"
  - **Standard Flow Enabled** = "On"
  - **Direct Access Grants Enabled** = "On"
  - **Backchannel Logout Session Required** = "On"

- Open ID Connect Compatibility Modes Settings
  - **Exclude Session State From Authentication Response** = "On"
  - **Use Refresh Tokens** = "Off"

Ensure that Access Type is set to confidential to get access to the Client Secret.

Open the `appsettings.json` (by default it can be found at `C:\Program Files (x86)\RayVentoryDataHub`).

Adjust the following entries to match the KeyCloak Client that will be used.

```
"ExternalIdentityManagement": {
   "IsEnabled": "true",
   "LoginUrl": "[YOUR_KEYCLOAK_INSTANCE/realms/[YOUR-REALM]/protocol/
openid-connect/auth?response_type=code&client_id=[YOUR-CLIENTID]",
   "ClientSecret": "[YOUR SECRET]",
   "Realm": "[YOUR REALM]",
   "Client": "[YOUR CLIENTID]",
   "AuthAPI": "[YOUR_KEYCLOAK_INSTANCE/realms/[YOUR-REALM]/protocol/
openid-connect/token"

},
```

An example default `appsettings.json` after a fresh install of RayVentory Data Hub can be found here.

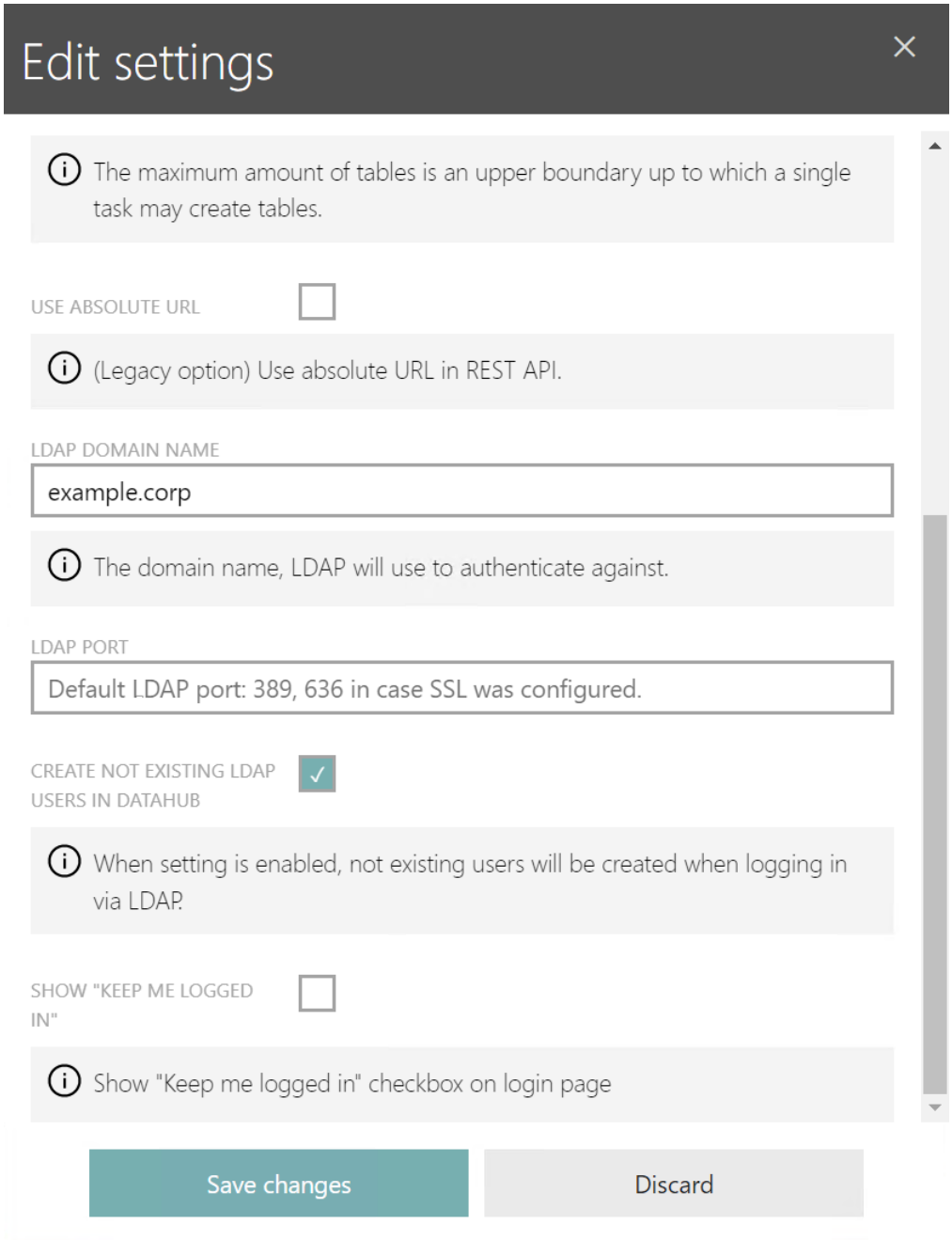
LDAP Configuration and Usage

In RayVentory Data Hub the Lightweight Directory Access Protocol (LDAP) is used to get user and group data from the Active Directory. This enables RayVentory Data Hub to synchronize AD users with RayVentory Data Hub users.

In order to use LDAP it first needs to be activated in the system settings. Without being activated LDAP features will not be available. In order to activate LDAP, go to **Site-Adminstration** > **System Settings**.

System Settings

In the **System Settings** tab, click on the **Edit** button located at the top left.

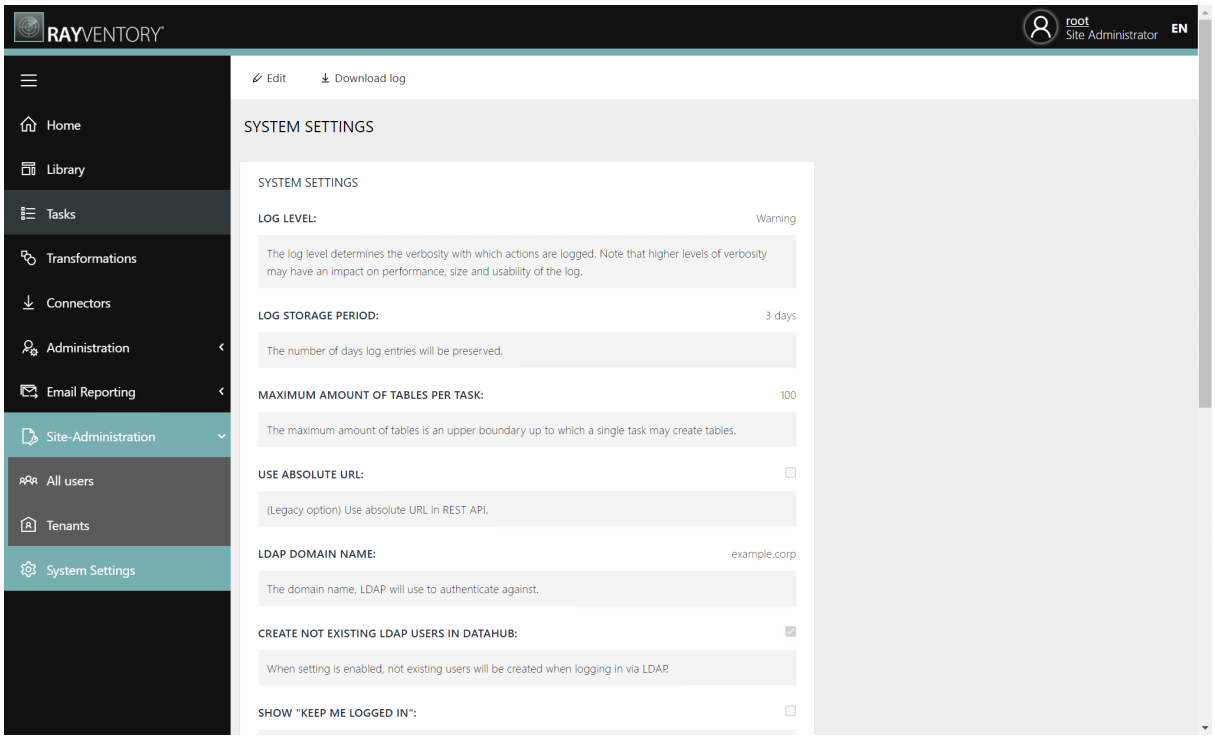


Enter the domain name into the **LDAP DOMAIN NAME** field.

If using a custom port, enter the port into the **LDAP PORT** field. If using one of the default ports, it is not necessary to enter a port into the field. The default ports used for LDAP are `389` for unsecured or STARTTLS connections and `636` for connections secured by TLS (LDAPS). Information on how to configure LDAP for TLS can be found in the LDAP Configuration Using the `appsettings.json` chapter.

In order to automatically create users that do not yet exist in RayVentory Data Hub, check the **CREATE NOT EXISTING LDAP USERS IN DATAHUB** checkbox. If the checkbox is checked, all LDAP users that log into RayVentory Data Hub will automatically be added to the RayVentory Data Hub user list.

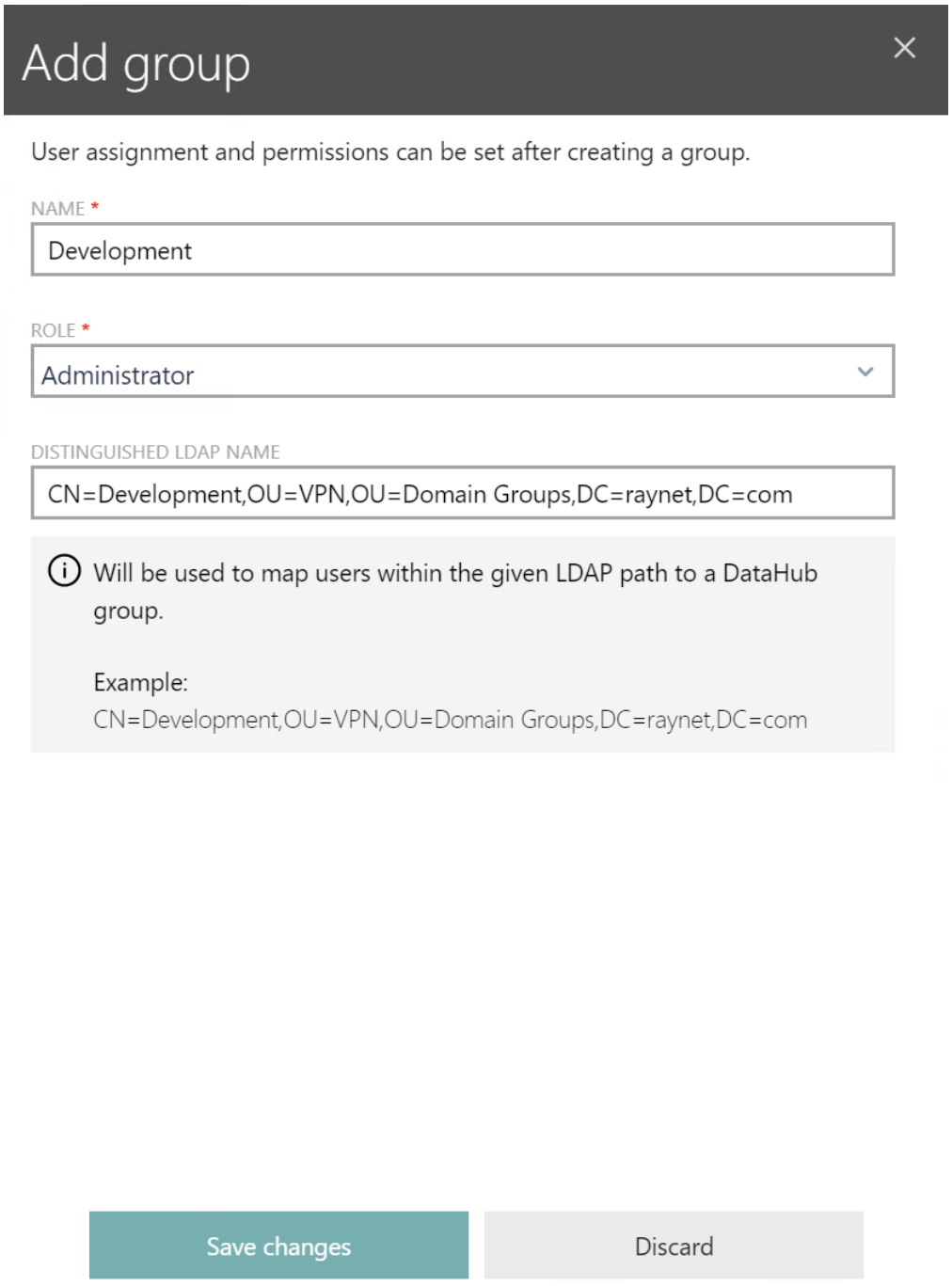
The LDAP settings (**LDAP DOMAIN NAME** and **CREATE NON EXISTING LDAP USERS IN DATAHUB**) will now be visible in the **System Settings** tab.



Groups

In the next step users and groups within the LDAP path can be mapped to a specific RayVentory Data Hub group. In order to do this, go to **Administration** > **Groups**.

Either click on the **+ Add** button to create a new group or go to the details page of an already existing group and click on the **Edit** button.



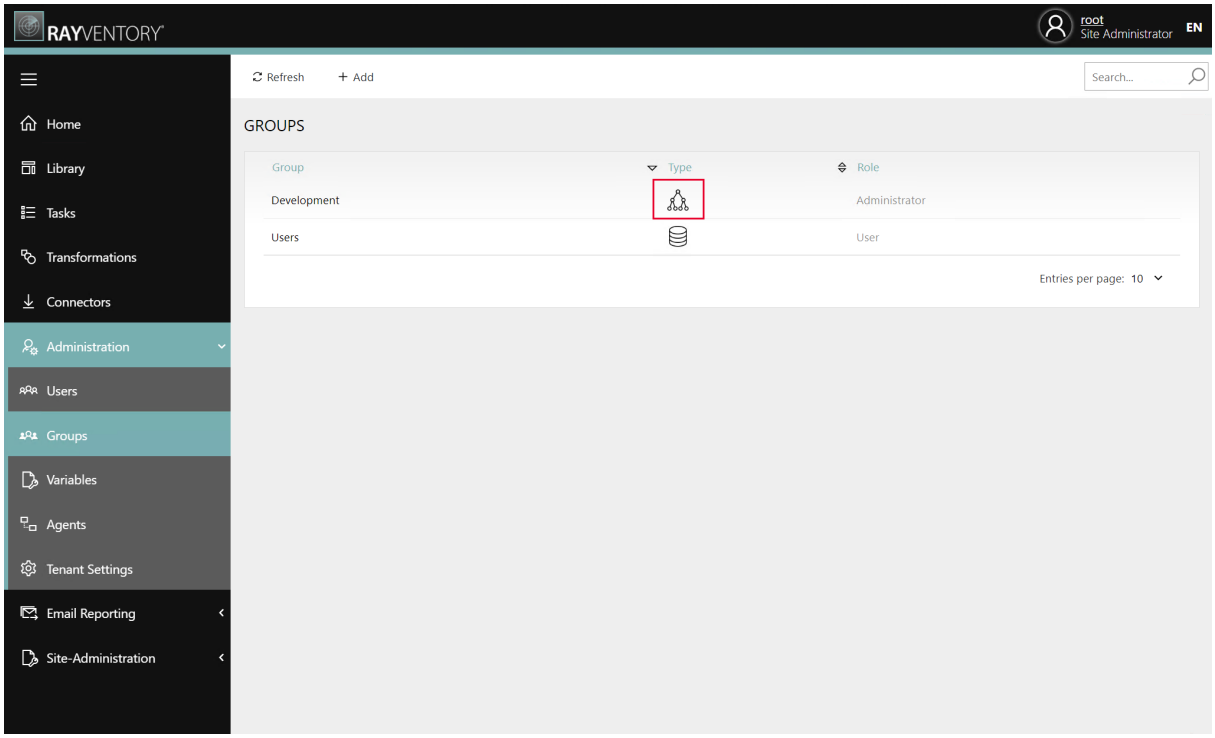
In order to map a group with users given within LDAP enter the Distinguished LDAP name into the **DISTINGUISHED LDAP NAME** field. The Distinguished LDAP name can be combined of any of the following attribute types.

String	Attribute Type
DC	domainComponent
CN	commonName
OU	organizationalUnitName
O	oragnizationName
STREET	streetAddress
L	localityName
C	countryName
UID	userid

All LDAP users that are mapped will become part of the RayVentory Data Hub group and will receive the rights that belong to users of that group.

Example:
CN=Development,OU=VPN,OU=Domain Groups,DC=raynet,DC=com

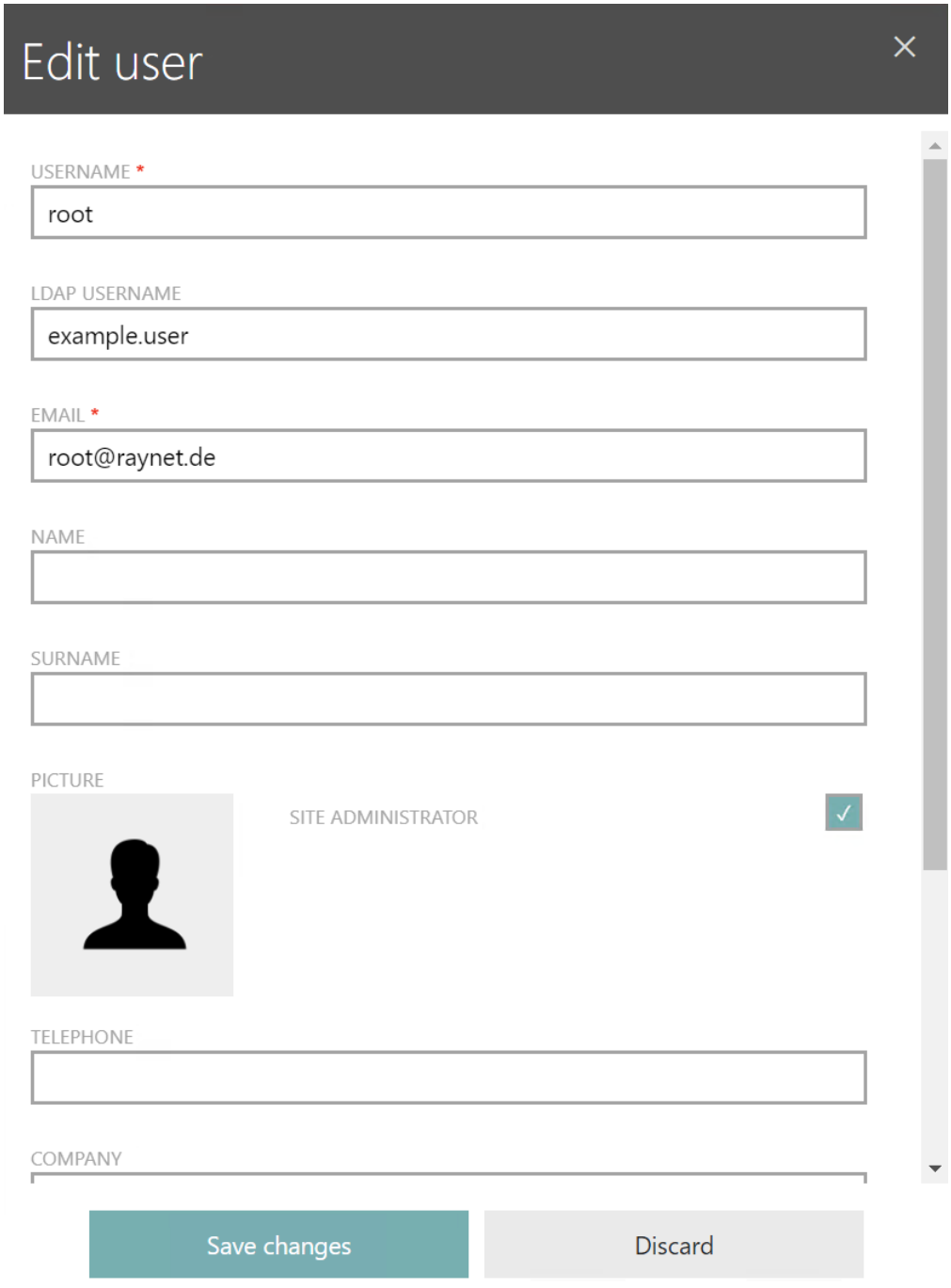
More information about distinguished names can be found in the Microsoft documenation.



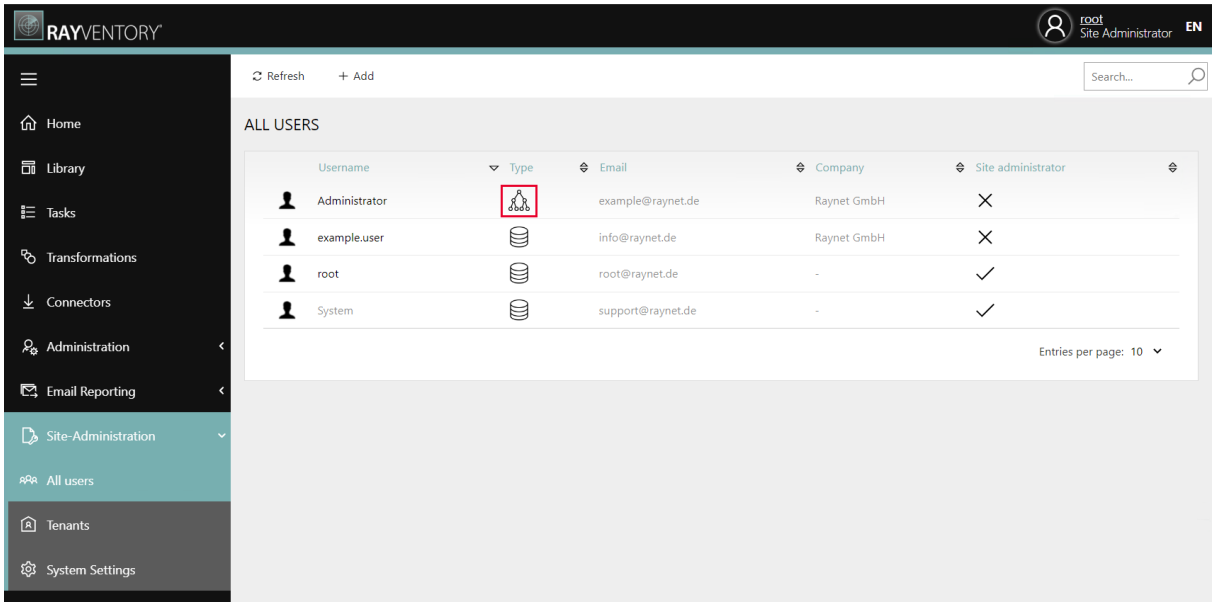
Groups that have been mapped with LDAP users will be shown with the following symbol in the **Type** column (the **Type** column is only visible if LDAP has been activated).

Connect a RayVentory Data Hub User with an Existing LDAP User

In order to map a specific user to an existing LDAP user, go to **Site-Adminstration** > **All users** and open the details page for the target user.



Enter the name of the LDAP user into the **LDAP USERNAME** field and save the changes. The user will now be mapped with the LDAP user. It is now possible to use either the credentials of the RayVentory Data Hub user or the credentials of the LDAP user in order to log in to this RayVentory Data Hub user account.

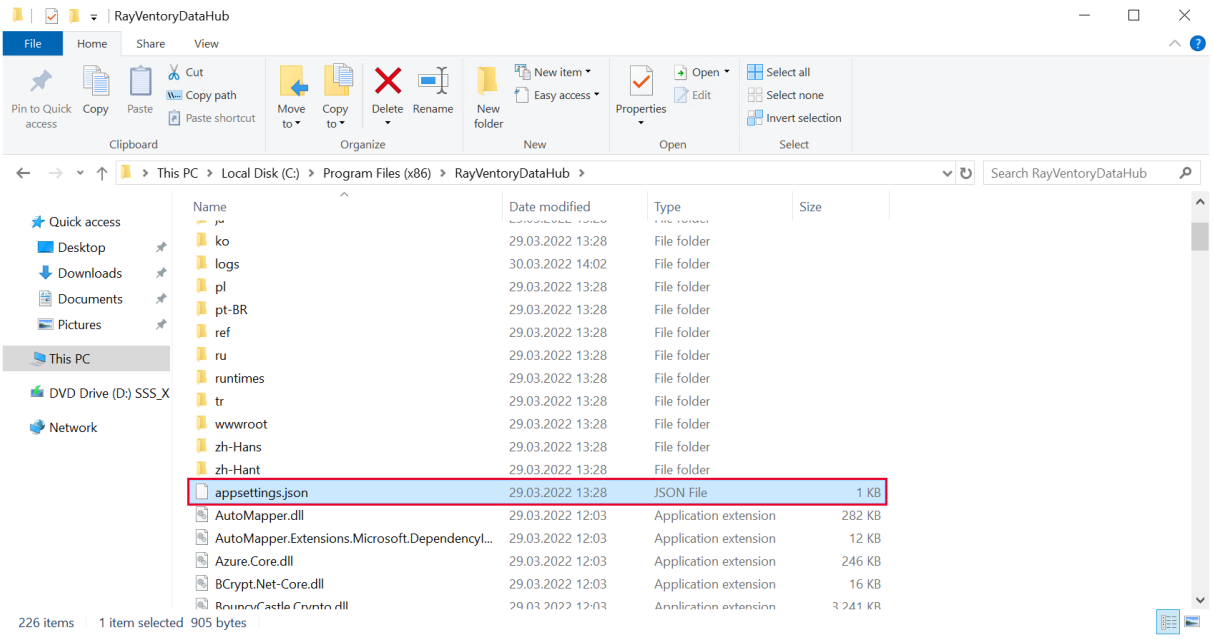


Users that have been mapped with LDAP users will be shown with the following symbol in the **Type** column (the **Type** column is only visible if LDAP has been activated).

Be aware:
Users that are linked to a directory service via LDAP cannot be set manually. Therefore it is no longer possible to remove them from a tenant. Any changes need to be done in the directory service.

LDAP Configuration Using the appsettings.json

It is possible to customize the LDAP feature using the appsettings.json file located in the [InstallDir] (by default C:\Program Files (x86)\RayVentoryDataHub) of RayVentory Data Hub.



Open the `appsettings.json` in an Editor of your choice. To find the LDAP settings search for `"LdapManagement"`.

Default `appsettings.json`

```
{
   "Logging": {
      "LogLevel": {
         "Default": "Trace"
      }
   },
   "TemporaryFilesDirectory":""
   "KotlinDirectoryPath": "",
   "TasksManagement": {
      "LogsDirectory": "",
      "DeleteLogFilesAfterDays": 30,
      "DeleteHistoryEntriesAfterDays": 90
   },
   "CsvFileProcessing": {
      "NumThreads": 3
   },
   "TaskDataTransformationProcessing": {
      "NumLocalThreads": 3,
      "CleanupThresholdInMb": 500
   },
   "TokenManagement": {
      "secret": "RayVentoryDataHubTopSecretSecret1337",
      "issuer": "Raynet GmbH",
      "audience": "RayVentory Data Hub User",
      "accessExpiration": 1,
      "refreshExpiration": 2,
      "rememberMeRefreshExpiration": 3
   },
   "reportViewerSettings": {
      "ReportDueTimeMinutes": 1,
      "ReportPeriodMinutes": 5,
      "ReportTimeToliveMinutes": 30,
      "ReportDocumentTimeToLive": 30,
      "ReportExportedDocumentTimeToLive": 30
   },
   "connectionStrings": {
      "System": "Server=192.168.125.193;Database=RayVentoryDataHub;User
   Id=raynet;Password=raynet123",
      "ReportDatabase": "Data Source=192.168.125.193;Initial
   Catalog=master;User Id=raynet;Password=raynet123"
   },
   "LdapManagement": {
      "Protocol": "LDAP",
      "Timeout": 5,
      "CustomUserFilter": "",
      "OpenLDAP_SASL_NOCANON": true,
      "OpenLDAP_AuthType": "Digest"
   },
   "ExternalIdentityManagement": {
      "IsEnabled": "false",
      "LoginUrl": "https://keycloack:8443/realms/RVDH/protocol/openid-
   connect/auth?response_type=code&client_id=rvdh-client",
```

```
    "ClientSecret": "XXXXXXXXXXXXXXXXXXXXXXXX",
    "Realm": "RVDH",
    "Client": "rvdh-client",
    "AuthAPI": "https://keycloack:8443/realms/RVDH/protocol/openid-
connect/token"

},
"AllowedHosts": "*"
}
```

The following table contains the settings for LDAP that can be found in the file.

Setting	Values/Range	Description
`Protocol`	A string containing one of the following values: • `LDAP` • `LDAPS`	Specifies if the connection that is used is unsecured or using STARTTLS or if the connection is secured by TLS. The default value is `LDAP`.
`Timeout`	Integer	Specifies the timeout in minutes. The default value is 5 minutes.
`CustomUserFilter`	A Unicode string as defined in RFC 2254. **Example:** `!(description=Trainee User*)`	Users defined by the filter will be restricted from access. By default the filter is empty. More information on LDAP search filters can be found in the Microsoft documentation.
`OpenLDAP_SASL_NOCANON`	Boolean (`true` or `false`)	This setting needs to be set to `true` in order to use LDAP with Linux. If it is set to `false`, LDAP cannot be used with Linux. By default, it is set to `true`. It is recommended to not change the setting, as it is only used with Linux.
`OpenLDAP_Authtype`	A string containing one of the following values: • `Unknown` • `Negotiate` • `Digest` • `External` • `ExternalAd` • `Simple` • `GssApi` • `Anonymus`	This is the authentication type that is used. By default, it is set to `Digest`.

Temporary File Path Configuration

RayVentory Data Hub uses temporary files when processing certain tasks and transformations. For tasks and transformations processed on the server, the location of the files and the threshold when to start the cleanup can be configured in the `appsettings.json`. The `appsettings.json` file located in the [InstallDir] (by default `C:\Program Files (x86)`

\RayVentoryDataHub) of RayVentory Data Hub Server. Information on how to configure this settings for tasks and transformations processed on the agent can be found here.

```
"TemporaryFilesDirectory":""
"CsvFileProcessing": {
   "NumThreads": 3
},
"TaskDataTransformationProcessing": {
   "NumLocalThreads": 3,
   "CleanupThresholdInMb": 500
```

Parameter	Required	Description
TemporaryFilesDirectory	{empty}	The TemporaryFilesDirectory specifies the directory where the server will write the temporary .csv files during the execution of the task or transformation. When the server has finished the collection of the data, it will send all written .csv files to the database and delete all temporary files in the directory. The TemporaryFilesDirectory needs to contain the path to the directory.
CleanupThresholdInMb	500	The CleanupThresholdInMb contains the threshold value in MB after which a cleanup of the temporary .csv files will automatically be started.

If no folder is defined, DataHub will automatically check for a temporary folder and select a folder according to the following rules:

Windows

- The path in the TMP environmental variable.
- The path in the TEMP environmental variable.
- The path in the USERPROFILE environmental variable.
- The Windows directory.

Linux

- The path in the TMPDIR environmental variable.
- If no path is defined, it chooses the /tmp/ defaultpath.

Advanced Topics

Data API

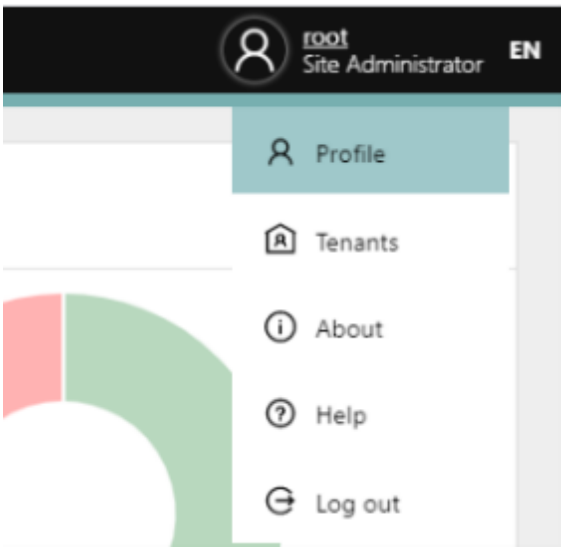
RayVentory Data Hub exposes a REST-based API, which can be used for automated query of various data sets available in the platform.

Authentication and Authorization

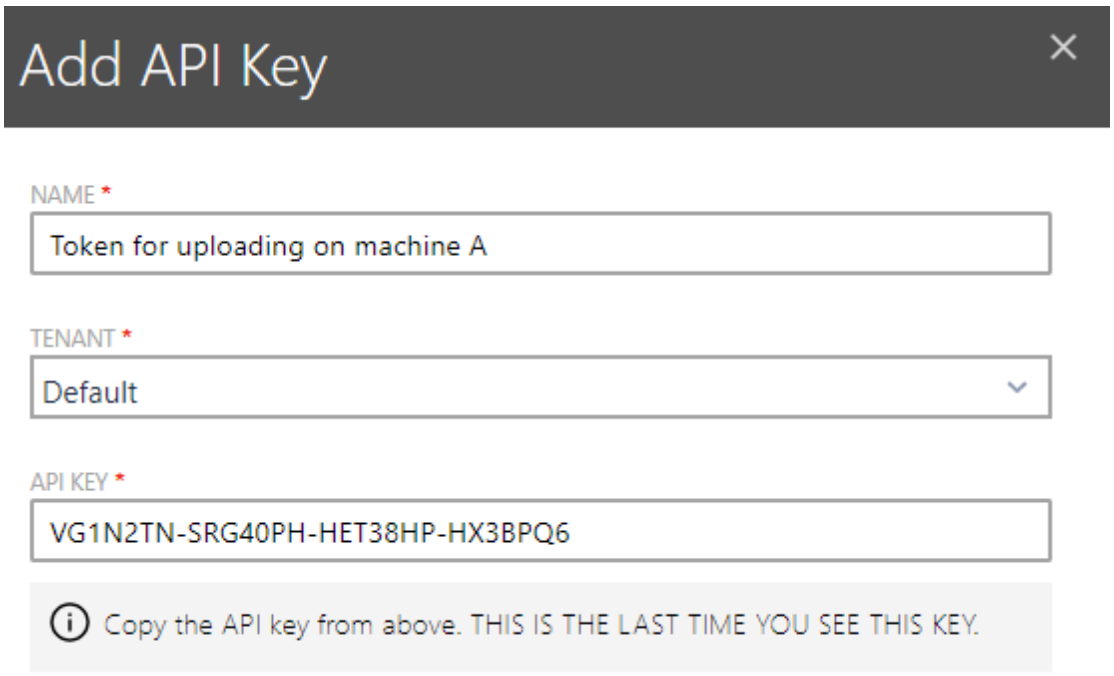
Every request requires a special token, called API key. By using a token, the underlying user to whom the token has been assigned will be impersonated, and the actions treated the same as if the user used his login and password.

To create a token:

1. Sign-in to RayVentory Data Hub.
2. Go to your **Profile** page.



3. Activate the tab **API KEYS**.
4. Press **+ Add** to open the token wizard.
5. Provide the required values.
   - **Name**: The display name of the token. This should be any value which describes the purpose or the owner of the token.
   - **Tenant:** A token is always valid for a specific tenant. This drop-down defines the assignment.
   - **API Key:** This is the actual API key used for authentication. You should leave the default value unless you want to brand your tokens with any pattern.



6. **Important:** Copy the token or write it down somewhere. As soon as you press **Save changes** or **Discard** this value will be not shown anymore!
7. Save changes by pressing **Save changes** button.

> **Note:**
> Tenant database can be optionally encrypted. The API endpoint ensures that the encryption is done transparently - the information sent to the endpoint should be always unencrypted, regardless of the backend and tenant settings.

Getting Available Table Names

This REST endpoint returns the list of available tables in JSON format.

Type of Request

GET

Endpoint URL

```
http://[host]:[port]/v1/resultDatabase/table
```

Query Parameters

This request has no configurable parameters.

Headers

Parameter	Required	Description
`ApiKey`	Yes	Your API key (see chapter Authentication and authorization for more information how to get it).

Sample (PowerShell)

The following code connects to the instance https://datahahub (using SSL and port 443) with authentication token `VG1N2TN-SRG40PH-HET38HP-HX3BPQ6` and then reads the names of available tables.

```
$dataHubApiKey = "M6KNS9Z-3404R00-Q42E4SG-1G4HKWT";
$dataHubHostName = "https://datahub.local";
$dataHubPort = 443;

$urlAddress = "{0}:{1}/v1/resultDatabase/table" -f
$dataHubHostName,$dataHubPort

$headers = @{};
$headers["ApiKey"] = $dataHubApiKey;

$request = Invoke-WebRequest -Uri $urlAddress -Headers $headers -Method
Get;

if ($request.StatusCode -eq 200)
{
    Write-Host "Available table names:";
    Write-Host (ConvertFrom-Json($request.Content));
}
else
{
    throw "Could not list the tables. HTTP code {0}" -f
$request.StatusCode;
}
```

This prints the following (output may be different depending on available tables)

```
Available table names:
Catalog_Software CustomTask MergedDevices
```

Getting Table Data (All Rows)

This is the simplest way of querying the data.

Type of Request

GET

Endpoint URL

```
http://[host]:[port]/v1/resultDatabase/table/<table-name>
```

Query Parameters

Parameter	Required	Description
`includeDataTypes`	No (default: false)	If set to true, then the CSV will receive an extra row after headers but before the data, which describe the actual column types (as indicated by the database engine). If you omit this parameter or set it to false, the returned CSV will not have this extra information, but will be then cross-compatible with other CSV-capable software.

Headers

Parameter	Required	Description
`ApiKey`	Yes	Your API key (see chapter Authentication and authorization for more information how to get it).

Sample (PowerShell)

The following code connects to the instance https://datahahub.local (using SSL and port 443) with authentication token `M6KNS9Z-3404R00-Q42E4SG-1G4HKWT` and then reads the content of the table **Catalog_Software**, which then gets written to local file `c:\temp\results.csv`.

```
$dataHubApiKey = "M6KNS9Z-3404R00-Q42E4SG-1G4HKWT";
$dataHubHostName = "https://datahub.local";
$dataHubPort = 443;
$tableName = "CatalogSoftware";
$outFile = "C:\temp\results.csv";

$urlAddress = "{0}:{1}/v1/resultDatabase/table/{2}" -f
$dataHubHostName,$dataHubPort,$tableName

$headers = @{};
$headers["ApiKey"] = $dataHubApiKey;

$request = Invoke-WebRequest -Uri $urlAddress -Headers $headers -Method
Get;

if ($request.StatusCode -eq 200)
{
    Write-Host $request.Content;
    $request.Content | Out-File $outFile
}
else
{
    throw "Could not list the tables. HTTP code {0}" -f
$request.StatusCode;
}
```

Getting Table Data (Paged Query)

This endpoints returns the data using paging. You can choose between the output format (CSV or JSON).

Type of Request

GET

Endpoint URL

`http://[host]:[port]/v1/resultDatabase/<table-name>/paged`

Query Parameters

Parameter	Required	Description
`page`	No	The number of the page to query. If omitted, the default value of 1 is taken (the first page).
`page_size`	No	The number of rows returned per-page. When omitted, the default value of 1000 items per page is used. The valid range for this parameter is 1-1000. Exceeding this range will coerce the value to the closest accepted value.
`includeDataTypes`	No (default: false)	This parameter is only relevant for CSV file requests. If set to true, then the CSV will receive an extra row after headers but before the data, which describe the actual column types (as indicated by the database engine). If you omit this parameter or set it to false, the returned CSV will not have this extra information, but will be then cross-compatible with other CSV-capable software.

Headers

Parameter	Required	Description
`ApiKey`	Yes	Your API key (see chapter Authentication and authorization for more information how to get it).

`Accept`	Yes	This determines the format of the data. Available supported values are:
		
		`application/json`
		`text/comma-separated-values`

Sample (PowerShell)

The following code connects to the instance https://datahahub.local (using SSL and port 443) with authentication token `M6KNS9Z-3404R00-Q42E4SG-1G4HKWT` and then reads the content of the table **CatalogSoftware**. It asks for the first page and returns 10 elements at once. Finally, it prints the column names with their types and the links for navigation:

```
$dataHubApiKey = "M6KNS9Z-3404R00-Q42E4SG-1G4HKWT";
$dataHubHostName = "https://datahub.local";
$dataHubPort = 443;
$tableName = "CatalogSoftware";
$pageSize = 10;
$page = 1;

$urlAddress = "{0}:{1}/v1/resultDatabase/table/{2}/paged?page={3}
&page_size={4}" -f $dataHubHostName,$dataHubPort,$tableName, $page,
$pageSize

$headers = @{};
$headers["ApiKey"] = $dataHubApiKey;
$headers["Accept"] = "application/json";

$request = Invoke-WebRequest -Uri $urlAddress -Headers $headers -Method
Get;

if ($request.StatusCode -eq 200)
{
    $parsedContent = ConvertFrom-Json($request);
    Write-Host "Available columns:";

    foreach ($item in $parsedContent.columns)
    {
```

```
        Write-Host (" * {0} ({1}) " -f $item.name, $item.type);
    }
    Write-Host "";

    Write-Host "Number of records:";
    Write-Host $parsedContent.records.Count;
    Write-Host "";

    Write-Host "Pages":
    Write-Host (" * First:    {0}" -f $parsedContent.pagination.first);
    Write-Host (" * Previous: {0}" -f
$parsedContent.pagination.previous);
    Write-Host (" * Next:     {0}" -f $parsedContent.pagination.next);
    Write-Host (" * Last:     {0}" -f $parsedContent.pagination.last);
}
else
{
    throw "Could not download the file. HTTP code $($req.StatusCode)";

}
```

This prints the following:

```
Available columns:
 * Id (String)
 * SoftwareId (String)
 * SoftwareVulnerabilityId (String)
 * Status (String)
 * VersionId (String)
 * ProductId (String)
 * Name (String)
 * Vendor (String)
 * RawVersion (String)
 * Version (String)
 * Architecture (String)
 * Language (String)
 * Website (String)
 * ProductFamily (String)
 * ParentFamily (String)
 * Functionality (String)
 * ReleaseDate (String)
 * EndOfLifeDate (String)
 * Support (String)
 * SoftwareType (String)
 * SoftwareClassification (String)
 * License (String)
 * AdditionalFunctions (String)
 * LatestReleaseDate (String)
 * LatestVersion (String)

Number of records:
10
```

```
Pages :
 * First:    https://datahub.local:443/v1/resultDatabase/resultTable/paged?
tableName=Catalog_Software&page=1&page_size=10
 * Previous:
 * Next:     https://datahub.local:443/v1/resultDatabase/resultTable/paged?
tableName=Catalog_Software&page=2&page_size=10
 * Last:     https://datahub.local:443/v1/resultDatabase/resultTable/paged?
tableName=Catalog_Software&page=233&page_size=10
```

Data Hub Agent

Data Hub Agent is a Windows-based agent which runs a Windows Service, that listens for queued tasks and executes them locally. The purpose of this component is also to transfer the collected data to the parent instance.

> **Note:**
> Data Hub Agent is required to perform automatic data collection from the RayVentory Data Hub UI.

The process of setting up a data agent has the following steps:

1. Identify the machine, from which scanned services are available and where there are enough permissions.
2. Check the prerequisites before installing the agent.
3. Install the agent.
4. Start the agent.
5. Authorize the agent.

All of these are described in details in the Installation and Configuration section.

Prerequisites

Supported Operating Systems

The following represents the list of supported operating systems at the time of release:

- Windows Vista SP2
- Windows 7 SP1
- Windows 8
- Windows 8.1
- Windows 10
- Windows 11
- Windows Server 2008 R2
- Windows Server 2008 SP1
- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

Hardware Prerequisites

- CPU: Intel Core i5
- RAM: 4GB
- Disk space: 500 MB

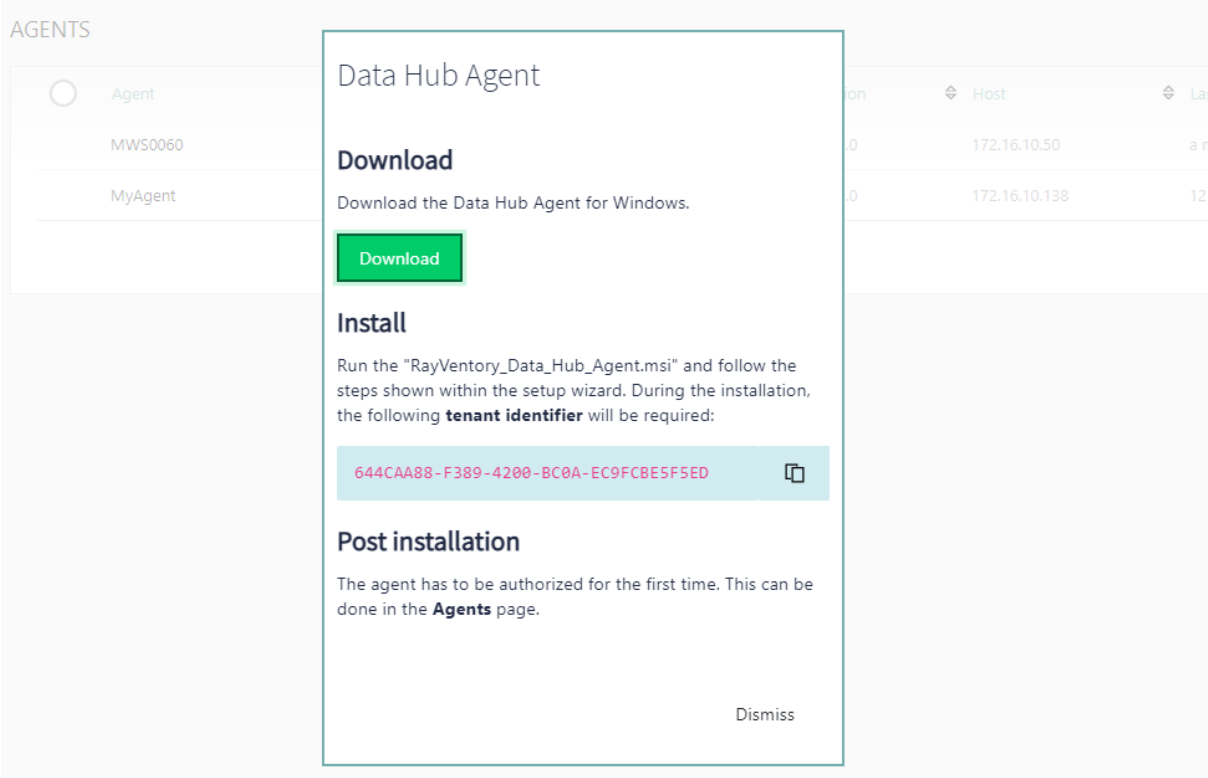
Software Prerequisites

- Microsoft .NET 6.0 – Windows Server Hosting Bundle (https://dotnet.microsoft.com/en-us/download/dotnet/6.0)
- Java / OpenJDK version 11 or newer is required to execute data collection from SaaS platforms.

Downloading the Data Hub Agent

Navigate to the **Agents** page under **Administration** using the navigation menu on the left panel. The list shows all agents which have been installed so far.

To install a new agent, press the **Install...** button. A pop-up with link and quick instructions will be shown:

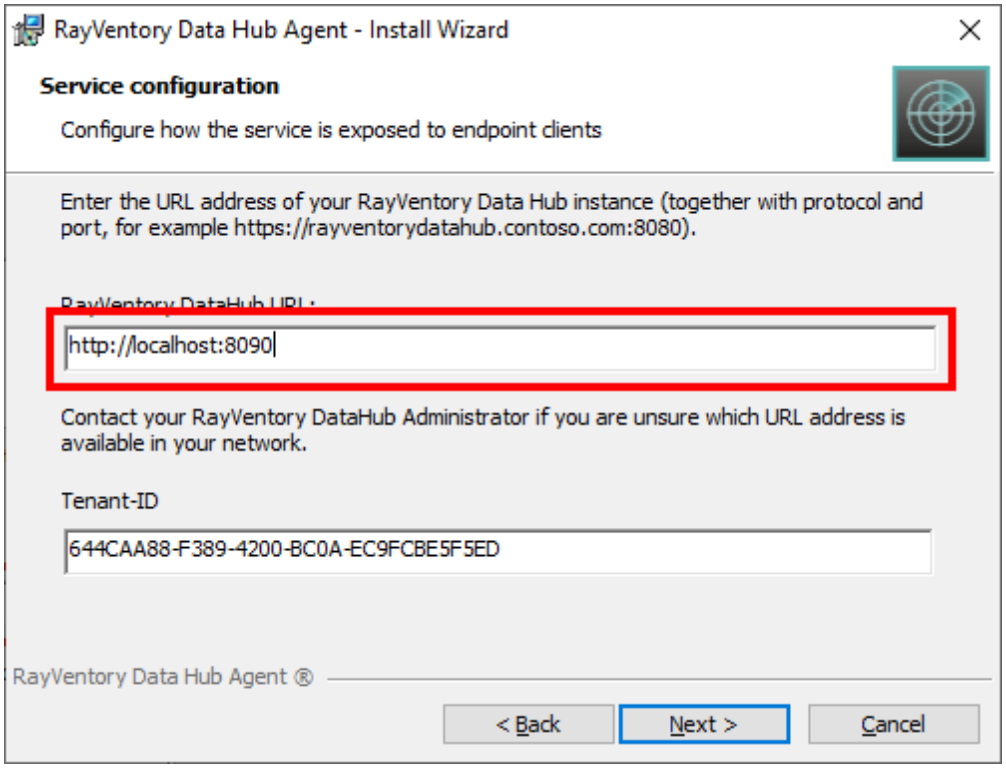


Installation

Run the "`RayVentory_Data_Hub_Agent.msi`" and follow the steps shown within the setup wizard. You will be asked for two important properties:

- The URL of RayVentory Data Hub - the full URL, together with the protocol and port number, for example `https://rayventorydatahub.local:8090`. When in doubt what the correct URL is, check out the address bar in your browser or ask your administrator.
- The tenant ID. This information is visible in the tenant selector, which is available from the login screen, settings or from the tenant switcher. You can also copy the tenant ID directly

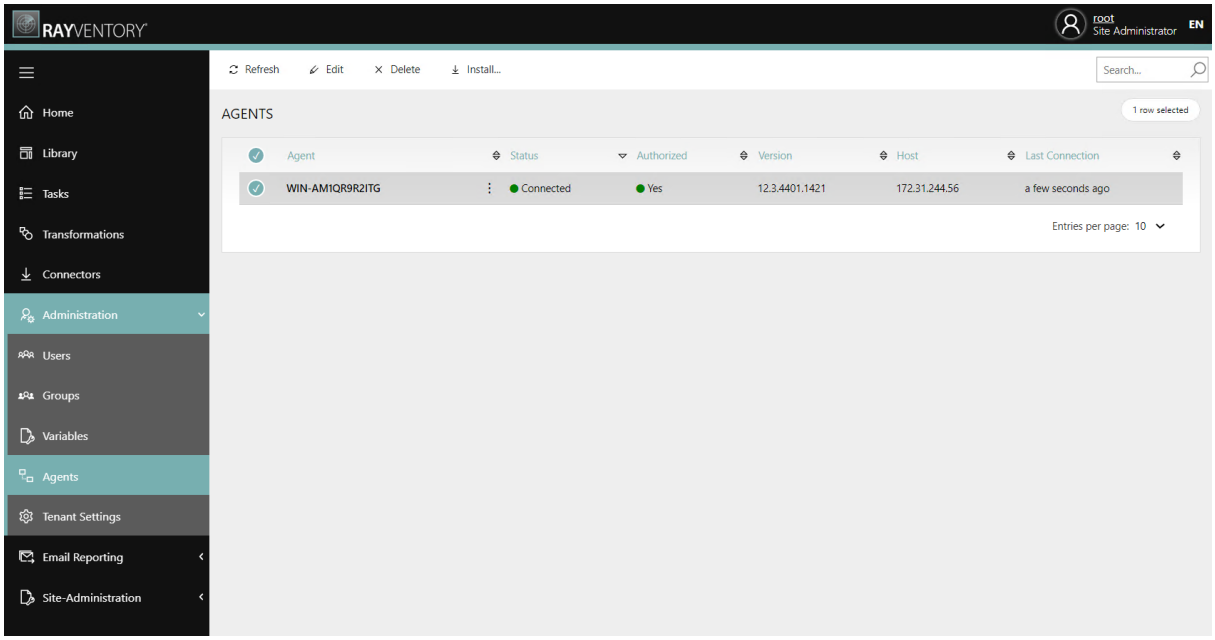
from the Download pop-up.



Registration

After installing the Data Hub Agent the agent automatically connects to the provided RayVentory Data Hub server URL.

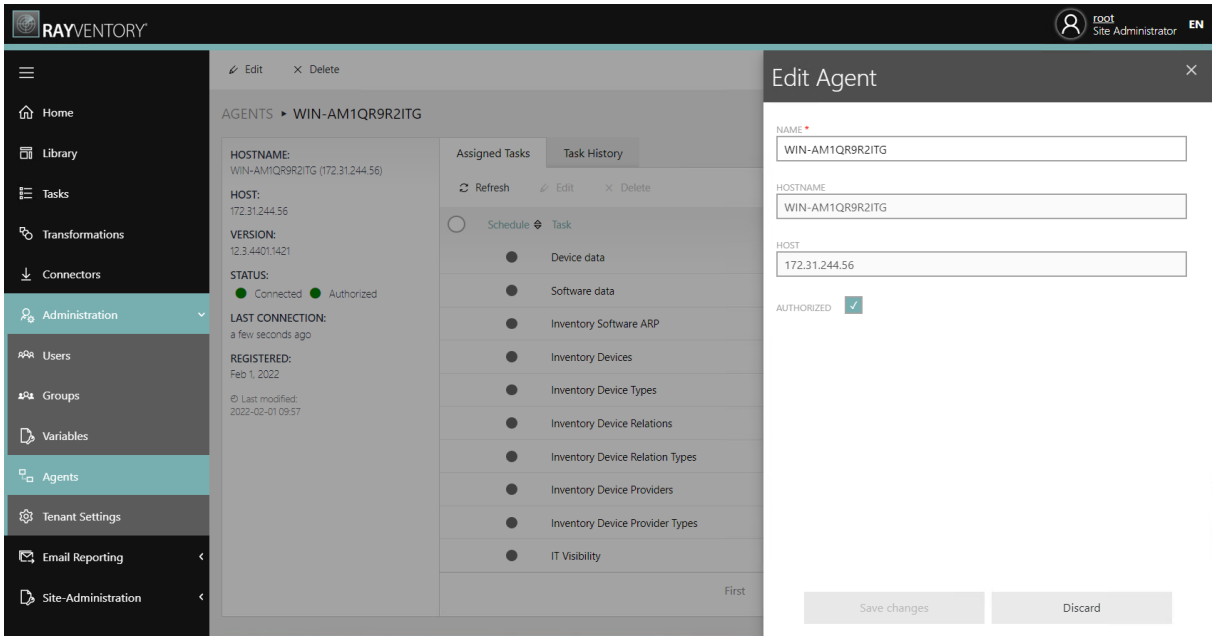
Visit the **Agents** page and search for an agent named after the machine the Agent has been installed on.



Note:
If the Data Hub Agent cannot be found, make sure that the **RayVentory Data Hub Agent** service is actively running on your machine. If it is not actively running, start it manually and visit the **Agents** page again.

Authorization

Visit the agent detail page by clicking on its name in the table. Edit its settings by clicking on “Edit” in the top action bar. A side panel appears to the right. Enable the "AUTHORIZE" check box in order to authorize the agent and save your changes. This agent is now authorized to request scheduled tasks that are assigned to it.



Next Steps

The agent is now ready to accept the tasks. You should now set-up necessary scheduled operations and the agent will listen to the changes and automatically pick up tasks from the pending queue.
See also the following chapter for more information about tasks: Tasks

Proxy Configuration

You can use the installer to configure basic proxy properties (host, port and credentials). If you install the agent on your own or a custom configuration is required, the changes can be done post-mortem via the configuration file.

The configuration is stored in file `Raynet.RayVentory.DataHub.Agent.dll.config` located in the installation folder.

The following XML Attributes can be set in the `<appSettings>` XML node:

Parameter	Required	Description
`ProxyHost`	Yes	The host of the proxy
`ProxyPort`	No	The port of the proxy
`ProxyUsername`	No	The user to be used to authenticate against the proxy

Parameter	Required	Description
`ProxyPassword`	No	The password to be used to authenticate against the proxy
`BypassProxyOnLocal`	No	A boolean value that indicates whether to bypass the proxy server for local addresses. true to bypass the proxy server for local addresses; otherwise, false. The default value is false.
`BypassList`	No	Set list of wildcards that describe URLs that do not use the proxy server when accessed - separated by a pipe character '|'. You can use the following special characters for matching: • `*` (asterisk) - matches zero or more characters • `?` (question mark) - matches exactly a single character

Temporary File Path Configuration

RayVentory Data Hub uses temporary files when processing certain tasks and transformations. For tasks and transformations processed on the agent, the location of the files and the threshold when to start the cleanup can be configured in the `Raynet.RayVentory.DataHub.Agent.dll.config`. The `Raynet.RayVentory.DataHub.Agent.dll.config` file located in the [InstallDir] (by default `C:\Program Files (x86)\RayVentory\DataHubAgent`) of RayVentory Data Hub Agent. Information on how to configure this settings for tasks and transformations processed on the server can be found here.

```
<add key="TemporaryFilesDirectory" value=""/>
<add key="ExecutionDelay" value="5"/>
<add key="Heartbeat" value="3"/>
<add key="ServerCertificateValidationCheck" value="false"/>
<add key="EtlCleanupThresholdInMb" value="500"/>
```

Parameter	Default Value	Description
`TemporaryFilesDirectory`	{empty}	The `TemporaryFilesDirectory` specifies the directory where the agent will write the temporary `.csv` files during the execution of the task or transformation. When the agent has finished the collection of the data, it will send all written `.csv` files to the backend and delete all files on the agent system. The `TemporaryFilesDirectory` needs to contain the path to the directory.
`EtlCleanupThresholdInMb`	500	The `EtlCleanupThresholdInMb` contains the threshold value in MB after which a cleanup of the temporary `.csv` files will automatically be started.

Windows

- The path in the `TMP` environmental variable.
- The path in the `TEMP` environmental variable.
- The path in the `USERPROFILE` environmental variable.
- The `Windows` directory.

Linux

- The path in the `TMPDIR` environmental variable.
- If no path is defined, it chooses the `/tmp/` defaultpath.

Logging

The log file of the RayVentory Data Hub Agent can be found in the following locations. These locations depend on the log-on user account of the agent service:

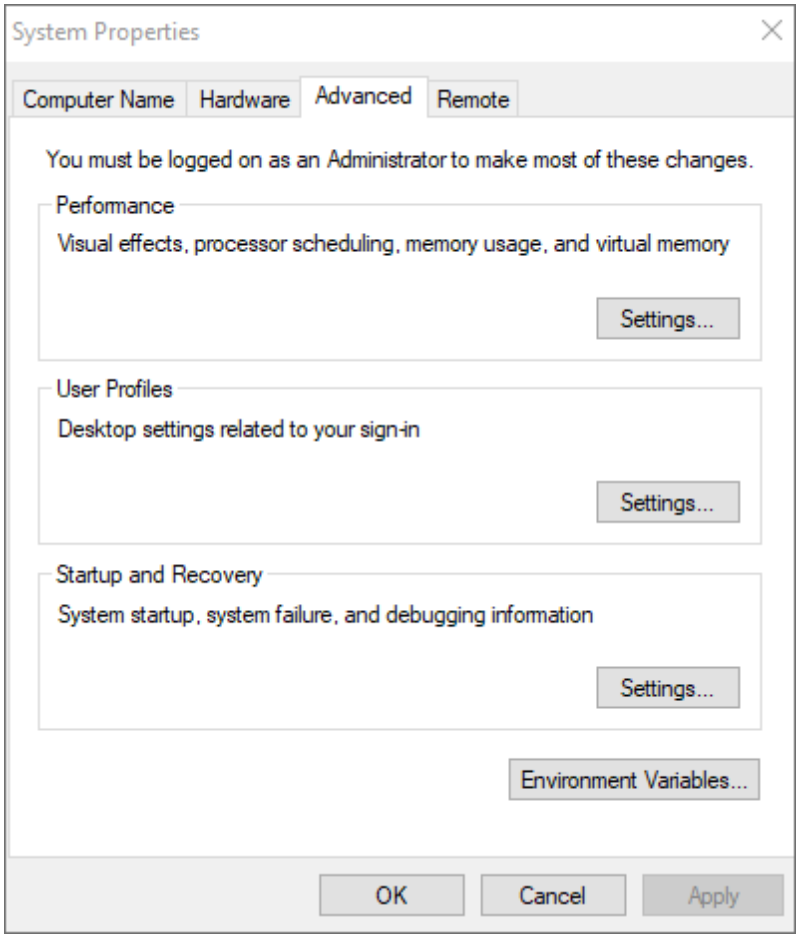
- **LocalSystem**: `C:\Windows\System32\config\systemprofile\AppData\Local\Raynet\RayVentory\Logs\Agent.log`
- **User account**: `C:\Users[USERNAME]\AppData\Local\Raynet\RayVentory\Logs\Agent.log`

Configuring the Java Environment Variable

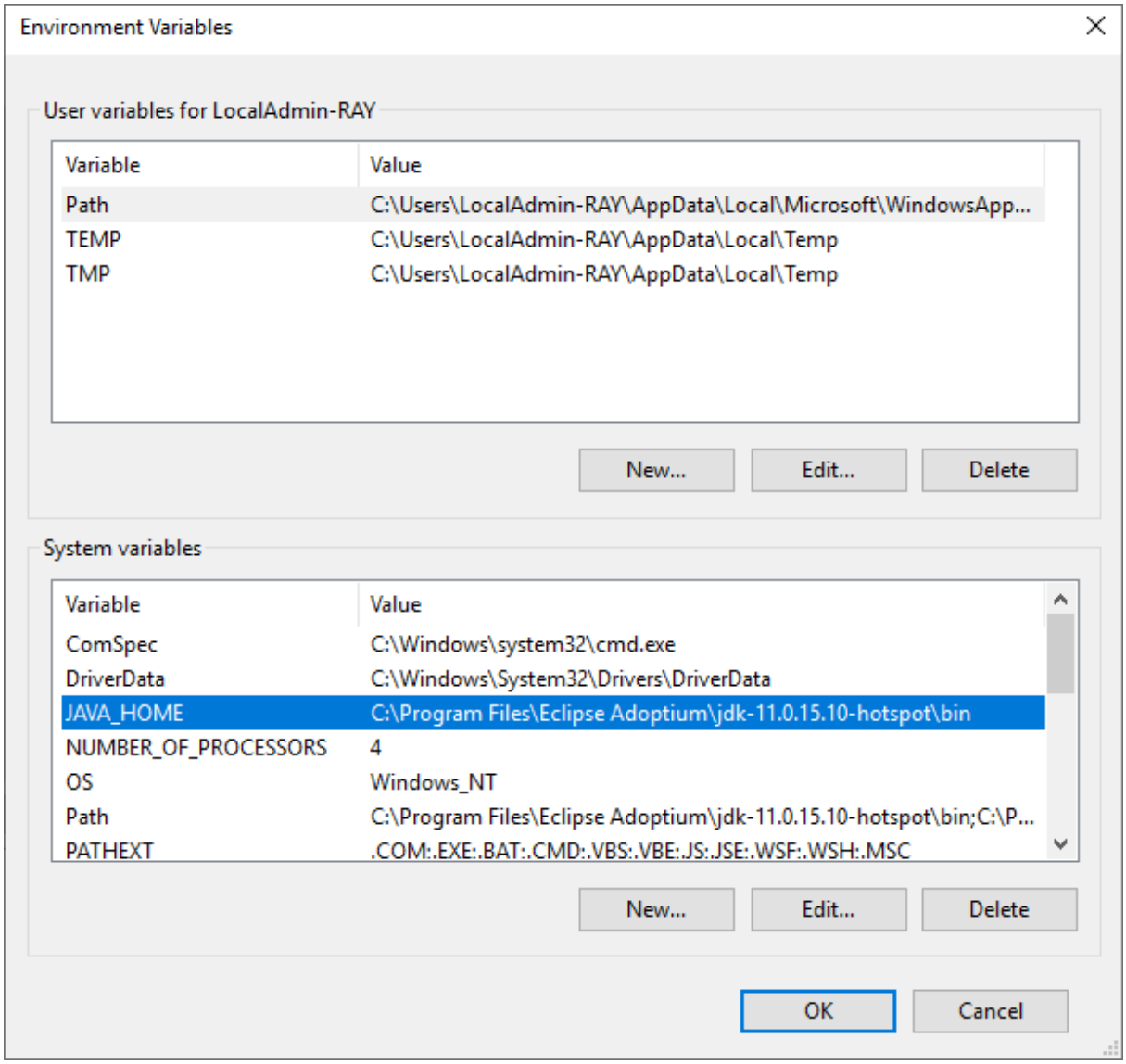
There are environments where it might be necessary to manually configure Java in order to be able to use the RayVentory Data Hub Agent. One use case where this might be necessary is, when multiple Java editions and/or versions are installed on a machine.

In order to configure a specific Java for RayVentory Data Hub if multiple Java versions are installed the **JAVA_HOME** enviroment variable needs to be configured manually.

Open the **System Properties** and select the **Advanced** tab.



In the **Advanced** tab click on the **Environment Variables...** button.



Search for the **JAVA_HOME** variable, click on the **Edit...** button and manually set the variable to the desired Java version.

Be aware:
The variable should be set with \bin at the end.

Be aware:
After installing a new Java or changing the environment variable the RayVentory Data Hub Agent service needs to be restarted to recognize the changes!

Appendices

Appendix I: List of Default Reports and Dashboards

RayVentory Data Hub comes with a number of default Reports and Dashboards. Information on how to setup RayVentory Data Hub in order to be available to access these default Reports and Dashboards refer to the Import Data from Template chapter. To find out which task creates the specific table used for a report or a dashboard refer to Appendix II: List of Tasks for the Default Reports and Dashboards.

Data Transformation

Path and Name	Type	Tables
Data transformation	Dashboard	• DataTransformation • ActiveDirectoryDevices • adobe_ac • vCenter • SCCMSQLDevicesCount • InventoryDevicesCount • Office365Data • ActiveDirectoryGroups • ActiveDirectoryUsers • aws_ec2 • aws_rds • SCCMSQLDeviceRelations • SCCMSQLSoftwareARP • SCCMSQLSoftwareFile • SCCMSQLSoftwareOS
Account data	Dashboard	• result_account_summary
Account data details	Report	• DataTransformation
Device data	Dashboard	• result_device_summary
Device data details	Report	• DataTransformation
Software data	Dashboard	• result_software_summary
Software data details	Report	• DataTransformation
User data	Dashboard	• result_user_summary
User data details	Report	• DataTransformation

Data Collection

Path and Name	Type	Datasources
ITSM connector	Dashboard	• CMP_BIOS • CMP_CASE • CMP_CDROM • CMP_DISK • CMP_FABRICKSWITCH • CMP_LANADAPTER • CMP_LOGICALDRIVE • CMP_MEMORY • CMP_MONITOR • CMP_MOTHERBOARD • CMP_NETWORK • CMP_OS • CMP_OS_FUNCTION • CMP_PRINTER • CMP_ROUTER • CMP_SOFTWARE • CMP_SYSSYSTEM • CMP_SYSTEM • CMP_TCP • CMP_USBDEVICE • CMP_VIDEO
RayVentory	Dashboard	• InventoryDevice • InventoryDeviceProviders • InventoryDevicyProviderTypes • InventoryDeviceRelations • InventoryDeviceRelationTypes • InventoryDeviceTypes • InventorySoftwareARP • InventorySoftwareFile • InventorySoftwareMicrosoft • InventorySoftwareMSI • InventorySoftwareOracle • InventorySoftwareOS • InventorySoftwareOthersw • InventorySoftwareTag • InventoryTerminalServer
ECM	Dashboard	• SCCMSQLCollections • SCCMSQLDeviceRelations • SCCMSQLDevices • SCCMSQLSoftwareARP • SCCMSQLSoftwareFile • SCCMSQLSoftwareFileMeter • SCCMSQLSoftwareOS • SCCMSQLSoftwareTag

Path and Name	Type	Datasources
		• sccm_wmi
Dynamics CRM	Dashboard	• dynamicscrm_org_data • dynamicscrm
AWS	Dashboard	• aws_ec2 • aws_rds
VMware vCenter	Dashboard	• vcenter
Citrix	Dashboard	• CitrixDirector • CitrixDirectorDeviceRelations • CitrixDirectorDevices
Active Directory	Dashboard	• ActiveDirectoryUsers • ActiveDirectoryDevices • ActiveDirectoryGroups
Office 365	Dashboard	• Office365Data
Azure Active Directory	Dashboard	• azuread
Adobe Admin Console	Dashboard	• adobe_ac • ActiveDirectoryUsers
Jira	Dashboard	• JiraUsers

Data Analysis

Path and Name	Type	Datasources
SaaS Management > **Dynamics CRM optimization**	Dashboard	• dynamicscrm • dynamicscrm-org_data
SaaS Management > **Jira User optimization**	Dashboard	• JiraUsers
SaaS Management > **Office 365 optimization**	Dashboard	• Office365Data
Hardware Asset Management > **Hypervisor overview**	Dashboard	• CustomSqlQuery
Hardware Asset Management > **Device software**	Report	• DataTransformation
Hardware Asset Management > **Hypervisor details**	Report	• DataTransformation

Path and Name	Type	Datasources
Hardware Asset Management > **Device details**	Report	• DataTransformation
Hardware Asset Management > **Devices**	Report	• DataTransformation
Hardware Asset Management > **Device hardware**	Report	• DataTransformation
Vulnerability Monitoring > **Vulnerability details**	Report	• VulnerabilityMonitoring
Vulnerability Monitoring > **Vulnerable software products**	Report	• VulnerabilityMonitoring
Vulnerability Monitoring > **Vulnerable device**	Report	• VulnerabilityMonitoring
Vulnerability Monitoring > **Vulnerability overview**	Dashboard	• VulnerabilityMonitoring • Vulnerability_MonitoringSummary
Portfolio Optimization > **Software portfolio overview**	Dashboard	• SoftwarePortfolioOverview • SoftwarePortfolioInformation • SoftwarePortfolioOverviewRSw
Portfolio Optimization > **Software on device**	Report	• Catalog
Portfolio Optimization > **Software portfolio details**	Report	• Catalog
Portfolio Optimization > **Software details**	Report	• Catalog
Software Asset Management > Database > Oracle > **Oracle DB Server Overview**	Report	• OracleServerOverview
Software Asset Management > Database > Oracle > Sub Reports > **Oracle DB Detail Overview**	Report	• HWInfoOS • InstanceLicenseInfo • LicenseSummary • Virtualization
Software Asset Management > Database > Oracle > Sub Reports > **Oracle DB Raynet Info**	Report	• OracleDFUSTable • OracleOption • rvsqlOraclevOptionTable • rvsqlOraclevParameter

Path and Name	Type	Datasources
Software Asset Management > Database > Oracle > Sub Reports > **Oracle DB DFUS Info**	Report	• `rvsqlOracleDFUSScriptData`
Software Asset Management > Database > Oracle > **Oracle DB Dashboard**	Dashboard	• `OracleDashboard`
Software Asset Management > **Installed software**	Report	• `DataTransformation`
Software Asset Management > **Software device**	Report	• `DataTransformation-`
Technology asset inventory	Dashboard	• `TechnologyAssetInventory`
OS > **OS Overview**	Dashboard	• `OperationSystem`
OS > **Devices by OS**	Dashboard	• `DevicesByOS`
IT visibility	Dashboard	• `ITVisibility` • `VulnerabilityInformation` • `HardwareOverview`

Appendix II: List of Tasks for the Default Reports and Dashboards

The tasks listed below are used by the default dashboards and reports. The setup of the tasks is described in the Import Data from Template chapter.

Path and Task	Table*	Connector
1. Data collection > Directory services > Active Directory > **Active Directory devices**	`ActiveDirectoryDevices`	Active Directory
1. Data collection > Directory Services > Active Directory > **Active Directory groups**	`ActiveDirectoryGroups`	Active Directory
1. Data collection > Directory Services > Active Directory > **Active Directory users**	`ActiveDirectoryUsers`	Active Directory
1. Data collection > Directory Services > Azure > **Azure AD**	`azuread`	Microsoft Azure AD
1. Data collection > IAAS - PAAS > AWS EC2 > **AWS EC2**	`aws_ec2`	Amazon Elastic Compute Cloud (EC2)
1. Data collection > IAAS - PAAS > AWS RDS > **AWS RDS**	`aws_rds`	Amazon Relational Database Services (RDS)
1. Data collection > Inventory > ECM SQL > **ECM SQL collection**	`SCCMSQLCollection`	Microsoft SQL Server
1. Data collection > Inventory > ECM SQL > **ECM SQL device relations**	`SCCMSQLDeviceRelations`	Microsoft SQL Server
1. Data collection > Inventory > ECM SQL > **ECM SQL devices**	`SCCMSQLDevices`	Microsoft SQL Server
1. Data collection > Inventory > ECM SQL > **ECM SQL software ARP**	`SCCMSQLSoftwareARP`	Microsoft SQL Server
1. Data collection > Inventory > ECM SQL > **ECM SQL software file**	`SCCMSQLSoftwareFile`	Microsoft SQL Server
1. Data collection > Inventory	`SCCMSQLSoftwareFileMetering`	Microsoft SQL Server

Path and Task	Table*	Connector
> ECM SQL > **ECM SQL software file metering**		
1. Data collection > Inventory > ECM SQL > **ECM SQL software OS**	SCCMSQLSoftwareOS	Microsoft SQL Server
1. Data collection > Inventory > ECM SQL > **ECM SQL software tag**	SCCMSQLSoftwareTag	Microsoft SQL Server
1. Data collection > Inventory > ECM SQL > **ECM SQL users**	SCCMSQLUsers	Microsoft SQL Server
1. Data collection > Inventory > ECM WMI > **ECM WMI**	sccm_wmi	SCCM (via WMI)
1. Data collection > Inventory > Hypervisor > Citrix > **Citrix Director**	CitrixDirector	Microsoft SQL Server
1. Data collection > Inventory > Hypervisor > Citrix > **Citrix Director device relations**	CitrixDirectorDeviceRelations	Microsoft SQL Server
1. Data collection > Inventory > Hypervisor > Citrix > **Citrix Director devices**	CitrixDirectorDevices	Microsoft SQL Server
1. Data collection > Inventory > Hypervisor > Vmware > **vCenter**	vcenter	VMware vCenter
1. Data collection > Inventory > RayVentory > **Inventory device**	InventoryDevice	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory device provider types**	InventoryDeviceProviderTypes	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory device providers**	InventoryDeviceProviders	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory device relation types**	InventoryDeviceRelationTypes	Microsoft SQL Server
1. Data collection > Inventory	InventoryDeviceRelations	Microsoft SQL Server

Path and Task	Table*	Connector
> RayVentory > **Inventory device relations**		
1. Data collection > Inventory > RayVentory > **Inventory device types**	InventoryDeviceTypes	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory software ARP**	InventorySoftwareARP	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory software file**	InventorySoftwareFile	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory software Microsoft**	InventorySoftwareMicrosoft	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory software MSI**	InventorySoftwareMSI	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory software Oracle**	InventorySoftwareOracle	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory software OS**	InventorySoftwareOS	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory software othersw**	InventorySoftwareOthersw	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory software tag**	InventorySoftwareTag	Microsoft SQL Server
1. Data collection > Inventory > RayVentory > **Inventory terminal server**	InventoryTerminalServer	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_BIOS**	CMP_BIOS	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_CASE**	CMP_CASE	Microsoft SQL Server
1. Data collection > ITSM >	CMP_CDROM	Microsoft SQL Server

Path and Task	Table*	Connector
EXPORT > **CMP_CDROM**		
1. Data collection > ITSM > EXPORT > **CMP_CPU**	CMP_CPU	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_DISK**	CMP_DISK	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_FABRICSWITCH**	CMP_FABRICSWITCH	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_LANADAPTER**	CMP_LANADAPTER	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_LOGICALDRIVE**	CMP_LOGICALDRIVE	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_MEMORY**	CMP_MEMORY	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_MONITOR**	CMP_MONITOR	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_MOTHERBOARD**	CMP_MOTHERBOARD	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_NETWORK**	CMP_NETWORK	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_OS**	CMP_OS	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_OS_FUNCTION**	CMP_OS_FUNCTION	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_PRINTER**	CMP_PRINTER	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_ROUTER**	CMP_ROUTER	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_SOFTWARE**	CMP_SOFTWARE	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_SYSSYSTEM**	CMP_SYSSYSTEM	Microsoft SQL Server

Path and Task	Table*	Connector
1. Data collection > ITSM > EXPORT > **CMP_SYSTEM**	`CMP_SYSTEM`	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_TCP**	`CMP_TCP`	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_USBDEVICE**	`CMP_USBDEVICE`	Microsoft SQL Server
1. Data collection > ITSM > EXPORT > **CMP_VIDEO**	`CMP_VIDEO`	Microsoft SQL Server
1. Data collection > ITSM > IMPORT > **Get PNP Matching Table**	`InventoryPNPMatchingTable`	Microsoft SQL Server
1. Data collection > ITSM > IMPORT > **RayVentory f_GetDeviceDiskDrive**	`InventoryDeviceDiskDrive`	Microsoft SQL Server
1. Data collection > ITSM > IMPORT > **RayVentory f_GetDeviceLogicalDisk**	`InventoryDeviceLogicalDisk`	Microsoft SQL Server
1. Data collection > ITSM > IMPORT > **RayVentory HardwareProperties**	`InventoryDeviceHardwareProperties`	Microsoft SQL Server
1. Data collection > ITSM > IMPORT > **RayVentory SNMP devices**	`InventorySNMPDevices`	Microsoft SQL Server
1. Data collection > ITSM > SaaS > **Adobe Connect**	`adobe_ac`	Adobe Connect
1. Data collection > ITSM > SaaS > **Dynamics CRM**	`dynamicscrm`	Microsoft Dynamics CRM
1. Data collection > ITSM > SaaS > **Dynamics CRM SKU prices**	`dynamicscrm_org_data`	Container
1. Data collection > ITSM > SaaS > **Jira users**	`JiraUsers`	Atlassian Jira (Cloud)
1. Data collection > ITSM > SaaS > **Office 365**	`Office365Data`	Microsoft 365
2.1 Data transformation > **Standardized Data Transformation**	`DataTransformation`	Data Transformation

Path and Task	Table*	Connector
2.2 Processing data transformation > **Account data**	`result_account_summary`	Microsoft SQL Server
2.2 Processing data transformation > **Device data**	`result_device_summary`	Microsoft SQL Server
2.2 Processing data transformation > **Software data**	`result_software_summary`	Microsoft SQL Server
2.2 Processing data transformation > **User data**	`result_user_summary`	Microsoft SQL Server
2.3 Data enrichment > **Catalog**	`Catalog`	RayVentory Catalog
3. Data analysis > Hardware Asset Management > **Device details**	`DeviceDetails`	Microsoft SQL Server
3. Data analysis > Hardware Asset Management > **Hypervisor details**	`HypervisorDetails`	Microsoft SQL Server
3. Data analysis > Hardware Asset Management > **Hypervisor overview**	`HypervisorOverview`	Microsoft SQL Server
3. Data analysis > IT visibility > **Hardware overview**	`HardwareOverview`	Microsoft SQL Server
3. Data analysis > IT visibility > **IT visibility**	`ITVisibility`	Microsoft SQL Server
3. Data analysis > IT visibility > **Vulnerability information**	`Vulnerability`	Microsoft SQL Server
3. Data analysis > OS > **Devices by OS**	`DevicesByOS`	Microsoft SQL Server
3. Data analysis > OS > **Operating System**	`OperatingSystem`	Microsoft SQL Server
3. Data analysis > Portfolio optimization > Software portfolio overview > **Software portfolio information**	`SoftwarePortfolioInformation`	Microsoft SQL Server

Path and Task	Table*	Connector
3. Data analysis > Portfolio optimization > Software portfolio overview > **Software portfolio overview**	SoftwarePortfolioOverview	Microsoft SQL Server
3. Data analysis > Portfolio optimization > Software portfolio overview > **Software portfolio overview redundant**	SoftwarePortfolioOverviewRSw	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Extract Oracle DFUS Script Data**	rvsqlOracleDFUSScriptData	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Extract Oracle DFUS Table**	rvsqlOracleDFUSTable	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Extract Oracle Instances**	rvspOracleInstances	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Extract Oracle Option**	rvsqlOracleOption	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Extract Oracle Option List Data**	rvsprpOracleOptionList	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Extract Oracle Report Data**	rvsprpOracleReport	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Extract Oracle Virtual Infrastructure**	rvsprpOracleVirtualInfrastructure	Microsoft SQL Server
3. Data analysis > Software Asset Management >	rvsqlOraclevOptionTable	Microsoft SQL Server

Path and Task	Table*	Connector
Database > Oracle > **Extract Oracle vOption Table**		
3. Data analysis > Software Asset Management > Database > Oracle > **Extract Oracle vParameter**	`rvsqlOraclevParameter`	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Transform Oracle Dashboard**	`OracleDashboard`	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Transform Oracle Server Overview**	`OracleServerOverview`	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Transform Oracle Sub Reports HWInfoOS**	`Oracle_sub_reports_HWInfoOS`	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Transform Oracle Sub Reports Instance License Info**	`Oracle_sub_reports_InstanceLicenseInfo`	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Transform Oracle Sub Reports License Summary**	`Oracle_sub_reports_LicenseSummary`	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Transform Oracle Sub Reports Oracle DFUS Table**	`Oracle_sub_reports_OracleDFUSTable`	Microsoft SQL Server
3. Data analysis > Software Asset Management > Database > Oracle > **Transform Oracle Sub Reports Oracle Option**	`Oracle_sub_reports_OracleOption`	Microsoft SQL Server

Path and Task	Table*	Connector
3. Data analysis > Vulnerability monitoring > **Vulnerability monitoring**	`VulnerabilityMonitoring`	Microsoft SQL Server
3. Data analysis > Vulnerability monitoring > **Vulnerability monitoring summary**	`VulnerabilityMonitoringSummary`	Microsoft SQL Server
3. Data analysis > **Technology Asset Inventory**	`TechnologyAssetInventory`	Microsoft SQL Server

* The names of the table in the list only represent part of the table names that will be created. The exact table names depend on the data that is delivered by a task. The information on how the exact table names are created can be found in the Naming Conventions chapter.

Appendix III: Default Configuration Files

This chapter contains the default versions of different files that are used for the configuration of RayVentory Data Hub.

appsettings.json

The `appsettings.json` file located in the [InstallDir] (by default `C:\Program Files (x86)\RayVentoryDataHub`) of RayVentory Data Hub Server.

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Trace"
    }
  },
  "TemporaryFilesDirectory": "",
  "KotlinDirectoryPath": "",
  "TasksManagement": {
    "LogsDirectory": "",
    "DeleteLogFilesAfterDays": 30;
    "DeleteHistoryEntriesAfterDays": 90
  },
  "CsvFileProcessing": {
    "NumThreads": 3
  },
  "TaskDataTransformationProcessing": {
    "NumLocalThreads": 3,
    "CleanupThresholdInMb": 500
  },
  "TokenManagement": {
    "secret": "RayVentoryDataHubTopSecretSecret1337",
```

```
    "issuer": "Raynet GmbH",
    "audience": "RayVentory Data Hub User",
    "accessExpiration": 30,
    "refreshExpiration": 720,
    "rememberMeRefreshExpiration": 10080
  },
  "reportViewerSettings": {
    "ReportDueTimeMinutes": 1,
    "ReportPeriodMinutes": 5,
    "ReportTimeToliveMinutes": 30,
    "ReportDocumentTimeToLive": 30,
    "ReportExportedDocumentTimeToLive": 30
  },
  "connectionStrings": {
    "Driver": "mssql",
    "System": "Server=192.168.125.151;Database=RayVentoryDataHub;
      User Id=raynet;Password=raynet123",
    "ReportDatabase": "Data Source=192.168.125.151;Initial Catalog=master;
      User Id=raynet;Password=raynet123"
  },
  "LdapManagement": {
    "Protocol": "LDAP",
    "Timeout": 5,
    "CustomUserFilter": "",
    "OpenLDAP_SASL_NOCANON": true,
    "OpenLDAP_AuthType": "Digest"
  },
  "ExternalIdentityManagement": {
    "IsEnabled": "false",
    "LoginUrl": "https://keycloack:8443/realms/RVDH/protocol/openid-connect
      /auth?response_type=code&client_id=rvdh-client",
    "ClientSecret": "XXXXXXXXXXXXXXXXXXXXXXXX",
    "Realm": "RVDH",
    "Client": "rvdh-client",
    "AuthAPI": "https://keycloack:8443/realms/RVDH/protocol/openid-connect/
token"
  },
  "AllowedHosts": "*"
}
```

Raynet.RayVentory.DataHub.Agent.dll.config

The `Raynet.RayVentory.DataHub.Agent.dll.config` file located in the [InstallDir] (by default `C:\Program Files (x86)\RayVentory\DataHubAgent`) of RayVentory Data Hub Agent.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
      <startup>
            <supportedRuntime version="v4.0"
sku=".NETFramework,Version=v4.5.2"/>
      </startup>
```

```
      <appSettings>
            <add key="DataHubUrl" value="http://localhost:8090"/>
            <add key="TenantId" value="2854F4AE-B863-453C-86B2-
FEEA25F51A9B"/>
            <add key="AgentId" value="{D616BA98-D060-4ECD-BA4D-
DC5A8653873A}"/>
            <add key="TemporaryFilesDirectory" value=""/>
            <add key="ExecutionDelay" value="5"/>
            <add key="Heartbeat" value="3"/>
            <add key="ServerCertificateValidationCheck" value="false"/>
            <add key="EtlCleanupThresholdInMb" value="500"/>
            <add key="ProxyUseHttp" value="true"/>
            <add key="ProxyUseHttps" value="true"/>
            <add key="ProxyUseSocks" value="false"/>
            <add key="ClientSettingsProvider.ServiceUri" value=""/>
      </appSettings>
      <runtime>
            <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
                  <dependentAssembly>
                        <assemblyIdentity name="Newtonsoft.Json"
publicKeyToken="30ad4fe6b2a6aeed" culture="neutral"/>
                        <bindingRedirect oldVersion="0.0.0.0-12.0.0.0"
newVersion="12.0.0.0"/>
                  </dependentAssembly>
            </assemblyBinding>
      </runtime>
</configuration>
```

Raynet GmbH
Technologiepark 22
33100 Paderborn, Germany
T +49 5251 54009-0
F +49 5251 54009-29
info@raynet.de

www.raynet.de