One Identity Manager 8.2

Operational Guide

One Identity Manager Operational Guide
Updated - 24 November 2021, 10:36
Version - 8.2

Contents

About this guide

The *One Identity Manager Operational Guide* provides an overview of the tasks and features that will be of assistance to you during normal operation of One Identity Manager.

This guide is intended for end users, system administrators, consultants, analysts, and any other IT professionals using the product.

NOTE: This guide describes One Identity Manager functionality available to the default user. It is possible that not all the functions described here are available to you. This depends on your system configuration and permissions.

The guide explains how to analyze and monitor changes to data in the Manager. It describes how you schedule activation times for operations. Basic tasks in One Identity Manager, such as editing schedules and mail templates as well as creating password policies, are explained. The guide also describes simple procedures that are used to export and import application data.

It explains how to declare changes to the configuration in the system, how to check data consistency and how to exchange custom changes between the development database, test database and productive database.

This guide does not describe the Operations Support Web Portal. For information about this topic, see the *One Identity Manager Operations Support Web Portal User Guide*.

Available documentation

You can access One Identity Manager documentation in the Manager and in the Designer by selecting the **Help** > **Search** menu item. The online version of One Identity Manager documentation is available in the Support portal under Technical Documentation. You will find videos with additional information at www.YouTube.com/OneIdentity.

Simulating data changes in the Manager

Using the simulation mode in the Manager, you can record and analyze the effects of comprehensive data changes to begin with before finally applying the changes.

The following information is recorded during the simulation:

- Calculation tasks for the DBQueue Processor resulting from the change
- Trigger changes that result from the change
- Processes that are generated as a result of the change
- Objects that are affected by the change
- Recalculations of compliance rules that result from the change

Detailed information about this topic

Prerequisites for using the simulation mode

- To use the simulation mode in the Manager, users need the **Option to start database simulation from the user interface** (Common_Simulation) program function.
- To re-calculate the compliance rules in simulation mode, enable the **Identity Audit Simulation** and **Identity audit simulation summary** plugins in the Manager

program settings.

- To ensure that the users can export the simulation data, enable the **Common | Simulation | ExportReport** configuration parameter in the Designer. If necessary, configure the report for exporting the simulation data.

Related topics

-

Configuring the simulation report

In the default One Identity Manager installation, the simulation report is created without the simulation data for evaluating the compliance rules.

To change the current report:

- In the Designer, in the **Common | Simulation | ExportReport** configuration parameter, enter the technical name of the report to be used to export the simulation data.

  Available report are:

  - **VID_DatabaseSimulationResult_Export**: The report shows the simulation data without evaluation of the compliance rules. This is the standard report.

  - **VID_DatabaseSimulationResult_with_Compliance_Export**: The report shows the simulation data including evaluation of the compliance rules.

Starting and ending simulation

⚠ **WARNING: You should only use the simulation mode in exceptional circumstances. During a simulation, the objects you are editing are locked for other users. Work on individual administration tools may be restricted. Under certain circumstances, the One Identity Manager Service stops running further processes during the simulation phase. Depending on the scope of the changes, the entire One Identity Manager environment can come to a standstill.**

NOTE:

- The active simulation mode is displayed in Manager by the symbol 📇 in the status bar and a red status bar displayed.

- To prevent an excessively long blockade of the overall system, simulation mode ends after 5 minutes if no data change is saved.

To run a simulation:

1. In the Manager, select **Database > Start simulation**.

2. Confirm the security prompt with **OK**.

   The program switches into simulation mode.

3. Make your desired changes.

4. To stop the simulation, click **Database > Stop simulation** in the Manager menu.

   The program switches to standard working mode and shows the simulation log.

   NOTE: After stopping the simulation, you can save the changes. In the Manager, select the **Object > Save** or **Object > Specify activation time** menu item.

Related topics

- Prerequisites for using the simulation mode on page 9
- Evaluating the simulation data on page 11
- Exporting simulation data on page 13

Evaluating the simulation data

When the simulation ends, the recorded modifications are loaded and displayed in the Manager as a protocol.

Table 1: Logging simulation data

View	Description	Displayed information	
Overview	This gives you an overview of which actions the applied changes will trigger. You can export the simulation data and display the report.	Number of applied changes for each action.	
DBQueue	The DBQueue log shows the following information. You can show the additional information from the context menu.	Operation	Calculation tasks to be run.
		Sort order	Sort order to process the calculation task.
		Process ID	Unique process ID.
		Object	Unique object ID.
		Child	Unique ID of the

View	Description	Displayed information	
		object	child object.
Generated process	Shows processes and process steps generated during simulation due to the changes. The individual properties of the processes and process steps are also displayed with their actual values.	Process	Name of the generated process.
		Process steps	Name of the generated process step.
		Property	Processes property or process step property.
		Value	Property value.
Trigger changes	Shows all changes made to objects that have been triggered during the simulation.	Table	Display text of the table to which the record belongs. This is used to group the objects.
		Object	Object affected by the change.
		Column	Column that was changed.
		Old value	Column value before the change.
		New value	Column value after the change.
Changed objects	Shows objects and their properties if they were affected by the changes made during simulation.	Table	Display text of the table to which the record belongs. This is used to group the objects.
		Object	Object affected by the change.
		Column	Column that was changed.
		Old value	Column value

View	Description	Displayed information	
			before the change.
		New value	Column value after the change.
Rule evaluation	During the simulation, the system recalculates all the rules that are affected by the changes. New rule violations and rule violations that no longer apply as a result of the recalculation are displayed.	Employee	Employee who violated the rule again or is no longer violating the rule for the first time.
		Rule violation	Type of change (rule no longer violated or new rule violation) and the affected rule.
		Description	Description of the rule violation.

Related topics

- Prerequisites for using the simulation mode on page 9
- Starting and ending simulation on page 10
- Exporting simulation data on page 13

Exporting simulation data

You can export the simulation log as a report in PDF format.

To export and display the simulation data as a report:

1. In the simulation log, select the **Overview** view.
2. Click the 🔲 button next to the list of actions.
3. Use the file browser to select the directory path for the report and enter a file name for the report.
4. To generate the .PDF file, click **Save**.
5. Click **Yes** to show the report now.

Related topics

-
-

Scheduling operations activation times

In addition, in the Manager, you can create, change, or delete an object at a specified point in time. The DBQueue Processor checks whether scheduled operations exist. When the scheduled time is reached, the operation is run by the One Identity Manager Service.

NOTE: As a rule, deferred deletion of user accounts by processes is defined in the table definitions as 30 days. After the deferred deletion limit has expired, the user accounts are deleted from the database

To run operations at a specified point in time

- In the Designer, check if the **Common | DeferredOperation** configuration parameter is set. Check the value of the configuration parameter and adjust it if necessary. Permitted values are:

  - **1** (default): In the Manager, you can schedule the activation time for creating, changing, or deleting an object.

  - **0**: Deferred deleted operation are carried out, such as, deferred deletion of user accounts. It is not possible to schedule the activation time in the Manager.

  You must recompile the database if you enable or disable the configuration parameter.

  If you disable the configuration parameter at a later date, model components and scripts that are not longer required, are disabled. SQL procedures and triggers are still carried out. For more information about the behavior of preprocessor relevant configuration parameters and conditional compiling, see the *One Identity Manager Configuration Guide*.

- In the Designer, check the **Common | DeferredOperation | AllowUpdateInInsertMode** configuration parameter and adapt it to the required behavior.

  - If this configuration parameter is disabled, an error occurs during processing if you try to insert an object that already exists in the database.

  - If this configuration parameter is enabled, when you insert an object that already exists in the database, the object is updated.

- In the Designer, check the **Common | DeferredOperation | IgnoreMissingOnDelete** configuration parameter and adapt it to the required behavior.

  - If this configuration parameter is disabled, an error occurs during processing if you try to delete an object that no longer exists in the database.

  - If this configuration parameter is enabled, missing objects are ignored during deletion.

Detailed information about this topic

- Planning activation times for actions in the Manager on page 16
- Displaying scheduled operations in the Manager on page 17

Planning activation times for actions in the Manager

In addition, in the Manager, you can create, change, or delete an object at a specified point in time.

To plan a activation time for creating and changing an object

1. In the Manager, select the object for which you wish to specify a activation time.
2. Select the **Change main data** task.
3. Change the values you wish to edit.
4. Select the **Object > Specify activation time** menu item.
5. Specify a change date.
6. Specify the time. To do this, select the hours or the minute display and change the setting using the arrow keys.
7. Enter additional information on the operation under **Remarks**.
8. Click **Save**.

To schedule a deletion time for an object

1. In the Manager, select the object for which you wish to schedule a deletion time.
2. Select the **Object > Set deletion time** menu item.
3. Specify the date and time of deletion.
4. Enter additional information on the operation under **Remarks**.
5. Click **Save**.
6. Confirm the security prompt with **Yes**.

Related topics

Displaying scheduled operations in the Manager

The scheduled operations with their activation times are displayed in an overview. If the scheduled activation time of an operation has passed or an error occurred when the operation ran, the corresponding entry is marked in red in the overview.

To display all scheduled operations

- In the Manager, click the **Database > Show deferred operations** menu item.

In the overview, the planned operation are displayed with the following information.

Table 2: Information on data changes

Information	Meaning
Table	Name of the table to which the data record belongs. This is used to group the objects.
Object	Object affected by the operation.
Operation	Operation that is run for the object. Permitted operations are **Add object**, **Change object**, **Delete object**, **Generate event**, and **Call method**.
Activation time	Time at which the operation should be run.
Comment	Additional comment on the operation. | TIP: Click a remark to show the remark in full.
Created by	User who created the scheduled operation.

Table 3: Meaning of icons in the form toolbar

Icon	Meaning
	Load and display the selected object.
	Run scheduled operations now
	Delete selected objects.
	Re-enable selected objects. If an error occurred during the operation, you can run

Icon	Meaning
	the change again.
↻	Reload the data.
▼	Filter view.

Related topics

- Restricting the display of scheduled operations on page 18

Restricting the display of scheduled operations

To limit the information for scheduled operations using defined filter conditions, use predefined filters. You can filter according to the statuses of the scheduled operations, or by scheduled operations.

To restrict the display

1. In the Manager, click the **Database > Show deferred operations** menu item.
2. In the overview of schedule operations, open the **Filter view** menu.
3. Select one or more filters under the **State** or the **Operation** item.

   TIP: To display all scheduled operations, go to the **Filter view** menu and select **Show all**.

Table 4: Predefined filters

Filter		Meaning
State	Outstanding operations	Shows or hides pending operations.
	Expired operations	Shows or hides operations whose activation time has already expired.
Operation	Create object	Shows or hides all entries with the **Add object** operation.
	Change object	Shows or hides all entries with the **Change object** operation.
	Delete object	Shows or hides all entries with the **Delete object** operation.
	Generate event	Shows or hides all entries with the **Generate event**

Filter	Meaning
	operation.
Calling methods	Shows or hides all entries with the **Call method** operation.
Show all	All scheduled operations are displayed.

Labeling input fields and assignments with planned changes

In the Manager, input fields and assignments with changes planned for a specific time, are labeled with additional icons. The new values are not shown for security reasons.

Table 5: Labeling of input fields and assignments with planned changes

Icon	Meaning
⚠️	The change in value is planned for a specific date and time. You can change the value only at the specified time.
📅	The change in value is planned for a specific date and time.

Re-applying templates

You can use templates in One Identity Manager to populate columns with default values or to map a column value from another column. For detailed information about templates, see the *One Identity Manager Configuration Guide*

In the Manager, you can re-apply the templates to the objects. This may be necessary if you have changed a template. In this case, column values determined by a template will be updated.

NOTE:

- Columns of an object are then also filled if they are not viewable on the current form in the Manager.
- This could also cause large numbers of dependent objects to be modified and processes to be generated.
- Templates defined in customizers are also run again.

To re-apply templates to the current object

1. In the Manager, select the object to which you wish to reapply the template.
2. Select the **Change main data** task.
3. In the menu, select the **Object > Reapply templates** menu item.
4. Save the changes.

Exporting data with the Manager

You can use the Manager to export application data. An export form in CSV format supports the data export. You can process the data further using a program such as Microsoft Office Excel or import it into another One Identity Manager database. You can export all data of a base table. In addition, you can export the data of tables that are linked by a foreign key relation to the base table.

You can save the export definition as a report that can be viewed and subscribed to in the Web Portal.

NOTE: To export data, users need the **Data export option** (`Common_DatabaseExport`) program function.

Detailed information about this topic

Creating data exports

Create an export definition and export the data to further process it with something such as Microsoft Office Excel or to import it into another One Identity Manager database.

NOTE: You can also export the file by selecting a menu item in the Manager navigation view. By default, the entries on the result list of the selected menu item are applied to the export. Under certain circumstances, the generated filter for the data set to be exported cannot be edited using the database query wizard. In this case, change the condition directly.

To create an export

1. In the Manager, select the **Database > Export data** menu item to open the export form.

2. In the **Column selection** pane in the **Base table** menu, select the table from which the data is exported.

   The database columns that can be exported are loaded and displayed in tabular form. The columns of the selected base table are displayed. In addition, all tables linked by a foreign key relation to the base table are displayed.

3. Select the columns that you wish to export and click the **Export** option.

   TIP: To mark all columns, use the ⊞ button in the toolbar. To clear all selected columns, click the ⊞ button. You can use the ⊠ button to display the display names or the technical names.

4. Use the **Export display value** option to set whether you wish to export actual values from the column or the display name. This may be necessary for database columns with special formatting, such as multilingual entries or a specified number of decimal places.

5. (Optional) In the **Columns to export** pane, use the ⬇, ↑, and ↓ buttons to adjust the sort order of the export columns.

6. (Optional) In the **Condition** pane, create a condition for further limiting the data records to be exported. The condition is defined as a valid where clause for database queries. You can enter the SQL query directly or with a wizard. Click 🪄 next to the field to open the wizard.

7. In the **Export data** pane, select the time zone for the export and create a preview of the export using the 🗔 button.

   NOTE: If you are exporting dates, such as the date of birth or leaving date, you can specify how blank dates are handled during export. To do this, use the button 🕒.

   The records that meet the export criteria are displayed in tabular form. Change how the data is sorted, if necessary. Click a column in the table header of the result list to sort by the selected column.

   NOTE: The sort order of the preview is not only used for display purposes, but also affects the data export. The data is exported as displayed in the preview.

8. In the **Export data** pane, use the 🖫 button to start the export. Use the file browser to select the directory path for the export and enter a file name for the export.

9. To generate the `.csv` file, click **Save**.

Related topics

Saving export definitions as simple reports

This creates a simple report using export definitions, which can be displayed and subscribed to in the Web Portal. You make this report available to Web Portal users.

> NOTE:
> - This function is only available if Report Subscription Module is installed.
> - To create a simple report with export definitions, enable the **Data export as report** plugin in the program settings in the Manager.
> - Simple reports that you create in the Manager can be displayed as statistics in the Manager's info system. To do this, you must alter the **Manager**'s user interface in the Designer. In the Manager's info system, the report opens when you double-click on the statistic's header.
>
>   For more information about how to implement statistics in the user interface for using in simple reports, see the *One Identity Manager Configuration Guide*.
> - You can define report parameters for simple reports. For detailed information, see the *One Identity Manager Report Subscriptions Administration Guide*.

To create a simple report with the export definition

1. In the Manager, select the **Database > Export data** menu item to open the export form.
2. Create the export.
3. Click in the title bar of the export form.
4. Enable **Simple list report**.
5. Click the button next to the report definition menu and enter the following information:
   - **Name:** Name of the report.
   - **Description:** Additional information about the report.
6. Click **OK**.
7. Click **Save**.

To make the report available to Web Portal users, assign the report to the employees.

For detailed information about editing reports and assigning them to employees, see the *One Identity Manager Report Subscriptions Administration Guide* and the *One Identity Manager Web Designer Web Portal User Guide*.

Related topics

- Saving export definition in user settings on page 25
- Saving export definitions in a file on page 24

- Creating data exports on page 21

Saving export definitions in a file

To make an export definition available to other users, save the export definition as a `.xml` file.

To save the export definition to a file:

1. In the Manager, select the **Database > Export data** menu item to open the export form.

2. Create the export.

3. Click 🖫 in the title bar of the export form.

4. Enable the **Save to file** option.

5. Open the file browser by pressing the button next to **Filename**, select the directory path and enter a name for the export definition.

6. Click **Save**.

   The `.xml` file is generated. The file browser is closed. The path and file name are displayed under **File name**.

7. Click **Save**.

To load an export definition from a file:

1. In the Manager, select the **Database > Export data** menu item to open the export form.

2. Click 📁 in the title bar of the export form.

3. Enable the **Load from file** option.

4. Open the file browser by pressing the button next to **Filename**, select the directory path and the file with the export definition.

5. Click **Open**.

   The `.xml` file is loaded. The file browser is closed. The path and file name are displayed under **File name**.

6. Click **Open**.

Related topics

Saving export definition in user settings

You can save an export definition in the user account configuration and reload it from there. If you store an export definition in the user account configuration, this export definition is only available to you.

To save an export definition to the user settings:

1. In the Manager, select the **Database > Export data** menu item to open the export form.

2. Create the export.

3. Click 🖫 in the title bar of the export form.

4. Enable the **Save in user settings** option.

5. Click the button 🔳 beside the **Export name** input field and enter a name for the export definition..

6. Click **Save**.

To load an export definition from the user settings;

1. In the Manager, select the **Database > Export data** menu item to open the export form.

2. Click 📁 in the title bar of the export form.

3. Enable the **Load from user settings** option.

4. Select the export definition from **Export name**.

5. Click **Open**.

To delete an export definition from the user settings:

1. In the Manager, select the **Database > Export data** menu item to open the export form.

2. Click 🖫 in the title bar of the export form.

3. Select **Save in user settings**.

4. Select the export definition from **Export name**.

5. Click 🔳 next to **Export name**.

6. To close the dialog, click **Cancel**.

Related topics

- Saving export definitions in a file on page 24
- Saving export definitions as simple reports on page 23
- Creating data exports on page 21

Analyzing data and data changes

In One Identity Manager, you can analyze data and data changes using different methods.

Reports

One Identity Manager provides several reports that present information about objects and their relations to other objects in the One Identity Manager database. For example, there are reports about employees and their user accounts, company structures, resources, and system entitlements, attestation, and compliance rule violations integrated into One Identity Manager. Identification, analysis, and summaries of relevant data are supported with the help of these reports.

The reports analyze data from both the One Identity Manager database and the One Identity Manager History Database. For more information, see Analyzing data changes in reports and the TimeTrace on page 28.

Use the Report Editor to create reports. For detailed information, see *One Identity Manager Configuration Guide*. You can view reports in the Manager. For more information, see Displaying reports in the Manager on page 27. Reports about system configuration are supplied in the Designer. For detailed information, see *One Identity Manager Configuration Guide*.

Report subscriptions

You can also send reports to specified email addresses using scheduled subscriptions. Web Portal users request subscribable reports and configure their own personal report subscriptions. The reports are delivered to Web Portal users by email as specified in a personally configured schedule.

For detailed information, see the *One Identity Manager Report Subscriptions Administration Guide* and the *One Identity Manager Web Designer Web Portal User Guide*.

TimeTrace

Use the TimeTrace function to track changes to an object that were made up to any point in the past.

In its analysis, the TimeTrace function includes the data changes saved to the One Identity Manager database as well as the records stored in a One Identity Manager History Database. You can use this to find out who had which permissions at which point in time.

You can apply historical data to the current object and restore the object to the status prior to the change. For more information, see Analyzing data changes in reports and the TimeTrace on page 28.

In the Manager, you can see the change data in the TimeTrace view. For more information, see Displaying change information in the Manager's TimeTrace on page 33.

Statistics

Statistics are recalculated at regular intervals and visualized in the user interface in various diagrams. This provides you with an overview of the system status at a glance.

In the Manager, you can see statistics in the Info system. For more information, see The Info system in the Manager on page 34. More statistics are available in the Web Portal. For detailed information about statistics in the Web Portal, see the *One Identity Manager Web Designer Web Portal User Guide*.

Displaying reports in the Manager

The Manager contains various reports about employees and their user accounts, company structures, resources and system entitlements, attestations and compliance rule violation evaluation.

To display a report in the Manager

1. Select the objManagerect you want to see the report for.

2. In the task view in the **Reports** section, select the report.

   This generates and displays the report.

TIP:

- Use the tooltip in the task view to show a more detailed description of the report.
- You can find additional report in the **My One Identity Manager** category.
- You can cancel the report while it is generating by clicking **Cancel** button.

In the report form you can change the window size and switch between pages. The following table shows other features.

Table 6: Functions for displaying reports

Button	Description	Shortcut
Print	Print report.	Ctrl + P
Save	Save report. There are additional save options on the selection menu.	
Send by email	Send report by email.	
Page size	Change page size.	Ctrl + Shift + S
Bookmarks	Set bookmark.	Ctrl + B
Parameters	Show the parameters applied when generating the report.	Ctrl + Enter
Find	Search in the report.	Ctrl + F
Full screen	Show full screen preview.	F2
Single page	Show a single page.	F3
Page width	Change the page width.	F5

Analyzing data changes in reports and the TimeTrace

In reports and in the TimeTrace, an analysis is carried out of the data changes saved in the One Identity Manager database and those stored in a One Identity Manager History Database. You can use this to find out who had what permissions at which point in time.

Figure 1: Analyzing data changes
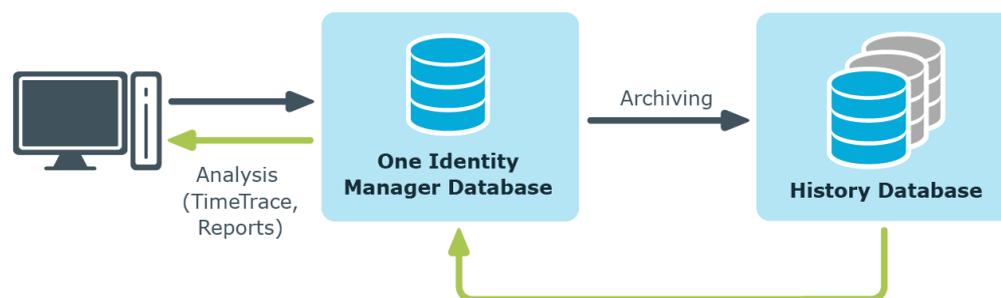
Prerequisite for analyzing historical data in reports and in the TimeTrace is the recording of data changes within process monitoring. Data changes that are saved in the One Identity Manager database, can be immediately included in the analysis. To use archived data, the One Identity Manager History Database must be declared in the One Identity Manager database.

Prerequisites for analyzing data changes

- The **Common | ProcessState** configuration parameter is set and a method for recording the data changes with process monitoring is configured.

  For more detailed information about logging changes to data, see the *One Identity Manager Configuration Guide*.

- To access archived data in the TimeTrace and in reports, the One Identity Manager History Database must be declared in the One Identity Manager database.

  There are different ways to establish a connection to the One Identity Manager History Database:

  - Method 1: Establish a connection to the One Identity Manager History Database through an application server.

    Use this method for accessing the One Identity Manager History Database over an encrypted connection. For more information, see Connecting a One Identity Manager History Database through an application server on page 30.

  - Method 2: Establish a direct connection to the One Identity Manager History Database.

    This method uses an unencrypted connection to access the One Identity Manager History Database. For more information, see Establishing a direct connection to a One Identity Manager History Database on page 32.

  For more information about archiving data, see the *One Identity Manager Data Archiving Administration Guide*.

- To display the TimeTrace view in the Manager, users require the **Option to show the TimeTrace** (Common_TimeTrace) program function.

- To evaluate of the effectiveness of the assignments (XIsInEffect column) in reports, in the Designer check the **Common | ProcessState | PropertyLog | ShowEffectiveAssignmentsOnly** configuration parameter and change it if necessary.

- To display only the effective assignments when evaluating historical assignments in reports, set the configuration parameter (default).
- To show all assignments as effective, regardless of their effectiveness, do not set the configuration parameter.

NOTE: The effectiveness of assignments (`XIsInEffect` column) is logged in the history as of One Identity Manager version 8.1.5. Older assignment data is always presented as effective, regardless of its actual effectiveness.

Related topics

Connecting a One Identity Manager History Database through an application server

Prerequisites for connecting a One Identity Manager History Database through an application server

- Declaring the One Identity Manager History Database in the TimeTrace, requires an ID.
- An ID for the One Identity Manager History Database connection is entered in the application server's configuration file (`web.config`).
    - Enter a unique ID for each One Identity Manager History Database.
    - The ID must be entered in all application servers that can be used by users to log in to the Manager.
    - The ID must be entered for the application server that the One Identity Manager Service uses to connect.
- The Manager and the Web Portal use the application server to log in. Otherwise, it is not possible to evaluate data modifications.
- To generate and send report subscriptions and reports by email that show changes to data, there must be a Job server set up over an application server.

For more information about setting up a Job server and about configuring the One Identity Manager Service, see the *One Identity Manager Configuration Guide*.

To link a One Identity Manager History Database into a TimeTrace

1. Use the Designer to log in to the One Identity Manager database.

2. In the Designer, select the **Base Data > General > TimeTrace databases** category.

3. Select the **Object > New** menu item.

4. Ensure that the **Use ID from application server** option is set.

5. In **History database name**, enter the name of the One Identity Manager History Database.

6. In the **Connection parameter** field, enter the ID for connecting to theOne Identity Manager History Database.

   The ID must match the ID in the application server's configuration file.

7. Select the **Database > Save to database** and click **Save**.

NOTE: Set the **Disabled** option to disable the connection at a later time. If a One Identity Manager History Database is disabled, it is not taken into account when determining change data in the TimeTrace.

To configure an ID in the application server for connecting to the One Identity Manager History Database

- During installation of the application server, enter the ID for connecting to the One Identity Manager History Database.

- To connect a One Identity Manager History Database at a later date, enter the ID for connection in the application server's configuration file (web.config) in the <connectionStrings> section.

Example:

```
<connectionStrings>

    ...

    <add name="<History Database ID>" connectionString="Data
    Source=<database server>;Initial Catalog=<database name>;User
    ID=<database user>;Password=<password>"/>

    ...

</connectionStrings>
```

NOTE:

The connection credentials in the application server's configuration file are encrypted with the default Microsoft ASP.NET encryption. If you want to change the connection credentials later, you must decrypt them first and then encrypt them again afterward. Use ASP.NET IIS registration tool to decrypt and encrypt (Aspnet_regiis.exe).

Example call:

Decrypting: aspnet_regiis.exe -pdf connectionStrings <path to web application in IIS>

Encrypting: aspnet_regiis.exe -pef connectionStrings <path to web application in IIS>

Related topics

- Establishing a direct connection to a One Identity Manager History Database on page 32

Establishing a direct connection to a One Identity Manager History Database

To link a One Identity Manager History Database into a TimeTrace

1. Use the Designer to log in to the One Identity Manager database.

2. In the Designer, select the **Base Data > General > TimeTrace databases** category.

3. Select the **Object > New** menu item.

4. Disable the **Use ID from application server**.

5. In **History database name**, enter the name of the One Identity Manager History Database.

6. Declare the **Connection parameters**.

   a. Click the [**...**] button next to the input field to open the input dialog for connection data.

   b. Enter the connection data for the One Identity Manager History Database.

      - **Server**: Database server.

      - (Optional) **Windows Authentication**: Specifies whether the integrated Windows authentication is used. This type of authentication is not recommended. If you decide to use it anyway, ensure that your environment supports Windows authentication.

      - **User**: SQL Server Login name of the installation user.

      - **Password**: Password for the installation user.

      - **Database**: Select the database.

7. Select the **Database > Save to database** and click **Save**.

NOTE: Set **Disabled** to disable the connection at a later time. If a One Identity Manager History Database is disabled, it is not taken into account when determining change data in the TimeTrace.

Related topics

-

Displaying change information in the Manager's TimeTrace

NOTE: To display the TimeTrace view in the Manager, users require the **Option to show the TimeTrace** (Common_TimeTrace) program function.

To display an object's change data:

1. In the Manager, open the time trace using the **View > TimeTrace** menu item.

2. Select the object whose change information you want to display.

3. Activate the change history for this object in the **TimeTrace** view using the 🕒 button.

4. In the **TimeTrace** view, use the ▼ (time range) filter in the toolbar to specify the time range for which the change information is loaded. The changes are determined from the One Identity Manager database and the connected One Identity Manager History Database databases.

   All change time stamps in the time frame that has been loaded are now shown in the overview below the timeline.

NOTE: To display changes of assignments to an object, such as the employee assignment to a department or a resource assignment to an organization, select the relevant assignment form in the task view of the Manager. In the **TimeTrace** view, you can then also select a source for which to display the changes. An additional **Source** menu is offered, in which you can select the respective assignment or the base object.

To select a change time stamp on the timeline:

- To display a part of the timeline in greater detail, click a marking below the timeline.

- Each change time stamp has a label showing the date and time. There is a tooltip for each change, showing which items of data were changed and by whom.

- Select a change time stamp on the timeline or on the label.

- If there are multiple change time stamps which are very close together, when you select a time stamp a context menu appears from which you can choose the specific change time stamp.

- Click the timeline or **Ctrl + mouse wheel** to zoom in or zoom on the display of several time change stamps that are close together.

When you select a change time stamp in TimeTrace, the program's document view opens the object's main data form or the assignment form. Use the timeline or quick edit a label

to choose if you want the object settings or assignments to be displayed in the main data form before or after the changes have been made.

If a property of an object shows a historical value, it is marked by an ⊚ icon. A tooltip shows the current value of the property. Use the **Show property change history** context menu to display the recorded data for this property.

You can apply historical data to the current object and restore the object to the status prior to the change.

To apply the historic values:

1. Click the ⊚ icon in front of the modified property. The following information is displayed.

    - **Property**: These properties are changed once the historical value is transferred. The changes are made immediately or by templates.

    - **New value**: Value of the property after the historical value has been saved.

    - **Old value**: Current value of the property. This value is overwritten once the historical value is saved.

2. Click **Save**.

The Info system in the Manager

The Manager's info system provides data about the health of the system in the form of diagrams.

- Topic-specific statistics and cross-functional statistics are displayed on the Manager home page.

- Within each category, topic-specific statistics are displayed under **Info system**.

- Topic-specific statistics for all categories are displayed in the **My One Identity Manager** category.

Statistics definitions form the basis of the info system. These are created centrally. For detailed information about creating statistics definitions, see the *One Identity Manager Configuration Guide*.

Every user can set which statistics they want to see and in which order. Use the settings to do this. The changes are saved to the user configuration so that the last setting used is shown when the program is restarted.

All the available statistics are grouped by topic in the settings. Each statistic has a title and a description.

To select statistics for display

1. Show the statistics and click **Settings**.

2. Enable or disable statistics by clicking the button in front of the name of the statistics.

   To restore the default setting, click **Restore default**.

3. Click **OK**.

Figure 2: Statistics settings
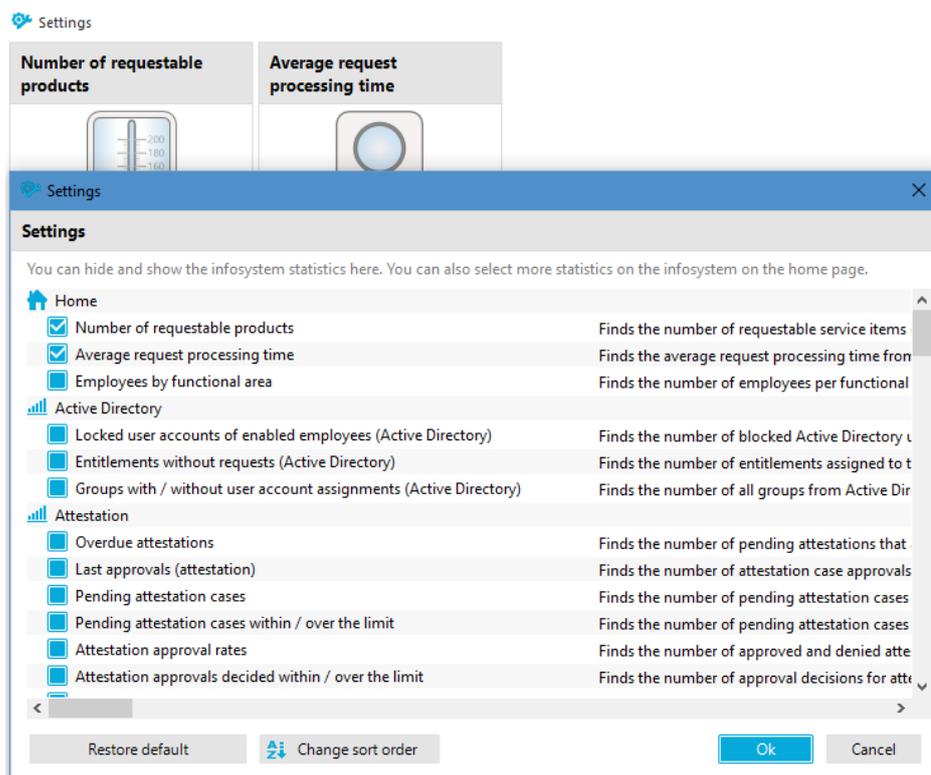
To change the order in which statistics are displayed:

1. Display the statistics and select **Settings**.

2. Click **Change sort order**.

3. Select the statistic that you wish to move. You can select multiple statistics using **Shift + select** or **Ctrl + select**.

4. Move the selected statistics with the arrow keys.

Table 7: Meaning of the buttons for changing the sort order.

Icon	Meaning
↑	Moves the selected statistics up.
↓	Moves the selected statistics down.

5. To restore the default setting, click **Restore default**.
6. Click **OK**.

   - OR -

   If you wish to make more changes, go back to the **Assignment view**.

Detailed information about this topic

Diagram types in the info system

There are several diagram types available for visualizing statistics.

Bar chart

A bar chart can be used to visualize comparisons between measurements.

Figure 3: Bar chart example
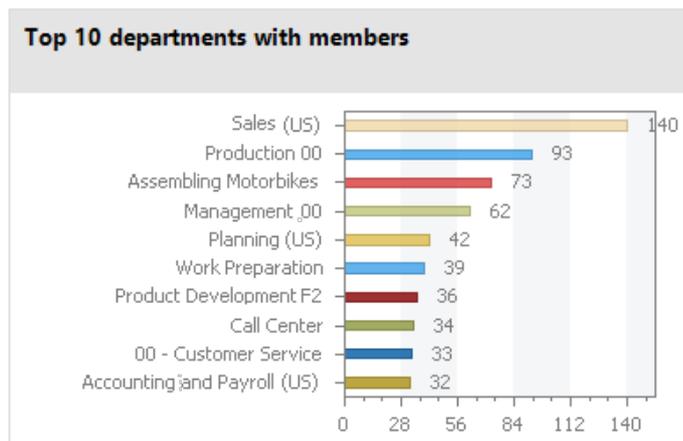
Pie chart

A pie chart can be used to visualize the measurements as a percentage of the base measurement.

Figure 4: Pie chart example
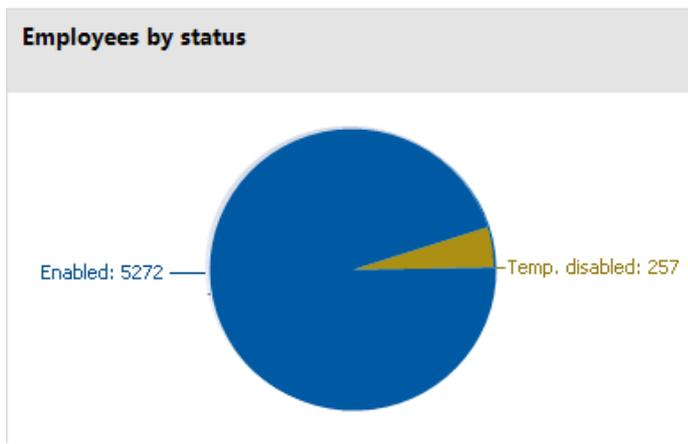
Line diagram

A line diagram can be used to visualize a data sequence over a specified time period. Click with the mouse on a point of measurement and a tooltip showing the measurement is displayed.

Figure 5: Line diagram example
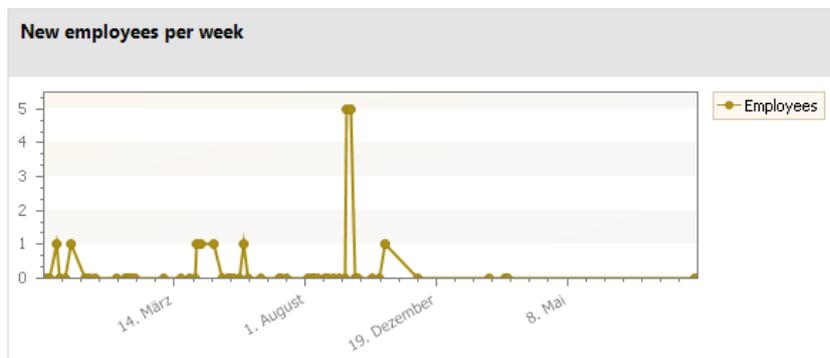
Traffic light

A traffic light diagram can be used to visualize the state of the system. The state is indicated by the color.

Table 8: Meaning of the colors

Color	State
Green	correct
Yellow	acceptable
Red	unacceptable

Figure 6: Traffic light example

Locked employees with
enabled user accounts

0
Employees
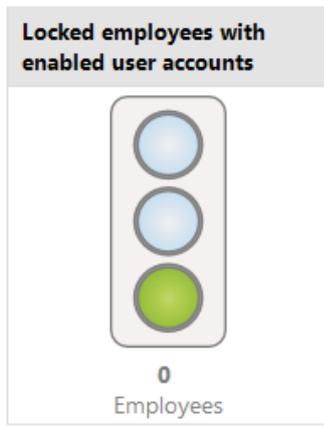
Tachometer

A tachometer diagram can be used to visualize the state of the system in more detail than in a traffic light diagram. The base measurement is also displayed. The state is indicated by the color.

Figure 7: Tachometer diagram example

Employees without
departments

915
Employees
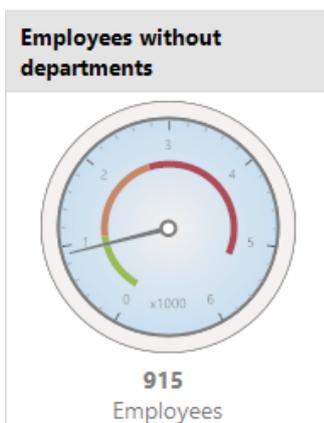
Thermometer

A thermometer diagram can be used to visualize the state of the system in more detail that in a traffic light diagram. The state is indicated by a color scale on the side of the diagram.

Figure 8: Thermometer diagram example

Employees without user accounts

5.206
Employees
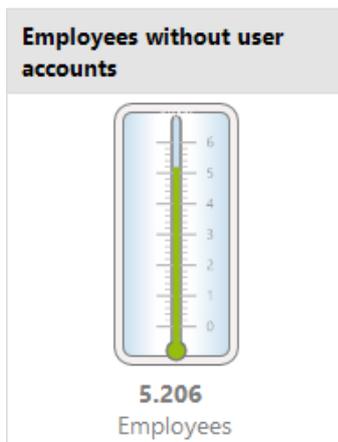
Table

Choose this diagram type to display the measurements or measurement trends over a certain timeframe in tabular format.

Figure 9: Table example

Number of employees		
	21.09.2017	
Employees	5.274	

Analyzing process monitoring in the Manager

In One Identity Manager, you have the option of logging the change history of objects and their properties. Different methods can be used to track changes within One Identity Manager.

For detailed information about the process monitoring methods, see the *One Identity Manager Configuration Guide*. For more information about configuring process monitoring of IT Shop requests, see the *One Identity Manager IT Shop Administration Guide*.

In the Manager's process view, the system shows the process data from running processes and process steps, the process data for direct database actions, and the recorded data changes in graphical format.

Detailed information about this topic

- Prerequisites for displaying the process information on page 40
- Working with the process view on page 41
- Opening the process view on page 41
- Process information layout on page 44
- Layout of logged data changes on page 45

Prerequisites for displaying the process information

- The process view in the Manager is only available if the **Common | ProcessState** configuration parameter is enabled and a method for monitoring the process is configured.
- The process view shows the process data only if the process data recording procedure is configured.

- The log is only displayed in the process view if the method for logging changes to data is configured and the logged in user has at least viewing permissions for the `Dialogwatch*` , `DialogProcess*`, and `QBMWatchOperationSummary` tables.
- To open the process view in the Manager, users need the **Option to show process information** (`Common_ProcessView`) program function.

For detailed information about configuring the process monitoring, see the *One Identity Manager Configuration Guide*.

Working with the process view

The process view is divided into two parts.

- The upper part of the process view displays a log containing the logged data changes. You can view the data changes of a process, a user and an object.
- The process information form is displayed in the lower part of the process view. This shows you an overview of the actions triggered in the system and the resulting processes. This displays information for the overall process and for the individual steps of a process.

You can configure the layout of process information. You can specify the level from which information is shown, for example, activities, details, or individual steps. You can restrict the scope of the information shown.

Related topics

- Prerequisites for displaying the process information on page 40
- Opening the process view on page 41
- Features in the process view on page 42
- Configuring process information on page 43
- Process information layout on page 44
- Layout of logged data changes on page 45

Opening the process view

To open the process view:

- In the Manager menu, select the **View > Process data** menu item.

Related topics

- Prerequisites for displaying the process information on page 40
- Features in the process view on page 42

Features in the process view

Table 9: Meaning of toolbar icons in the process data form

Icon	Meaning
↻	Reload process data.
👤	Show process data for the current user (object-related process data).
⊞	Show process data for the selected object (object-related process data).
⇦	Show processes for related objects.
⇨	Show substitute processes.
▼	Filter process data by status.
📇	Show data changes for the current user in the log (user-specific changes).
📇	Show data changes to the object selected in the result list in the log (object-related changes).

Table 10: Items in the process data form context menu

Context Menu Item	Meaning
Search	The system searches for objects in the process view.
Add to favorites	Adds the selected object to your favorites.
Remove from favorites	Removes the selected object from your favorites.
Tasks	The object's available forms are shown and you can switch to the desired form.
Object type:<BaseObject>	This shows the base object of the triggered process
Show process logs	The log shows the data changes of the selected process (process-related changes).
Properties	Show other properties of the active object. This menu item is only available in expert mode.

Table 11: Meaning of toolbar icons in the log

Icon	Meaning
🔍	The selected object appears in the document view.
🔧	The display switches to the originally referenced (old) object and this is shown in document view.
🔧	The display switches to the newly referenced object and this is shown in document view.

Configuring process information

To configure the process display in the Manager

1. In the Manager, select the **Database > Settings** menu item.

2. On the **Functionality** tab, configure the following settings in the **Process information** pane.

   - **Display complexity**: Set the display range. Permitted values are:
     - **Activities**: Activity information (top hierarchy level) is shown.
     - **Details**: Information about activities and their details is shown.
     - **Single steps**: Information about activities, details, and individual steps at the selected depth is shown.

   - **Single step details**: Set the depth of detailed information shown for individual steps. Permitted values are:
     - **Basic information**: Individual steps with a detail depth of **basic information** are shown.
     - **Extended information**: Single steps with a detail depth of **basic information** and **extended information** are shown.
     - **Complete information**: Single steps with a detail depth of **Basic information**, **Supplementary information**, and **Complete information** are shown (technical view).

   - **Show whole tree**: If this option is activated, the entire hierarchy tree automatically opens when the process view is loading. If this option is deactivated, the hierarchy tree is not opened when the process view is loaded.

   - **Show selected process automatically**: If this option is activated, the entire hierarchy tree automatically opened when a process is selected. If this option is deactivated, the hierarchy tree is not opened when a process is selected.

Process information layout

The process information form in the process view provides you with an overview of the actions triggered in the system and the resulting processes. This displays information for the overall process and for the individual steps of a process.

To display the recorded process data:

- To show all the current user's processes, click 👤 (user-specific) in the process data form.
- To show all of an object's processes, select the object in the result list and click ⊞ (object-specific) in the process data form.

The following process data appears:

Table 12: Logged process data

Information	Meaning
Activities	Process data display text for the process.
Status	Process status.
Triggered by	User who triggered the process.
Triggered on	Time of action.
Duration	Processing time.
More inform-ation	More information on the status, such as attempts to repeat individual steps or a start time for deferred steps.
Process ID	Unique ID (`GenProcID`). Changes that can be traced back to a single cause are given the same Process ID and are grouped in this way.
	TIP: To copy a process ID, click to select the process ID and copy the process ID to the clipboard using **Ctrl + C**.

The following icons are used to identify process statuses:

Table 13: Meaning of the icons for the process statuses

Icon	Meaning
✅	Processing was completed with success (status **Finished**).
▶	The process is currently being processed (status **Active**).
❌	An error occurred during processing (status **Error**).
‖	Status of processing (status **Pending**, **Delayed**, **Frozen** or **Not reached**).
✚	Process dependent on selected process.

Icon	Meaning
◄	Previous substitute process.
►	Next substitute process.

Related topics

- Layout of logged data changes on page 45

Layout of logged data changes

Individual data changes to the process view are displayed in the document view in the form of a log.

To show recorded data changes:

- To show all data changes that were run within a process, select the process in the process data form and click **Show logs for this process** in the context menu.
- To show all data changes carried out by the current user, click 🖳 in the process data form.
- To show all of an object's data changes, select the object in the result list and click 🖳 in the process data form.

The data changes log shows the following information.

Table 14: Information on data changes

Information	Meaning
Change history	This shows the affected object and the changed properties. To give a better overview, objects are grouped according to the table to which the dataset belongs.
Change date	Time of action.
Changed by	User who made the changes.
Old value	Column value before the change.
New value	Column value after the change.

Table 15: Meaning of icons in the log

Icon	Meaning
▦	Column

Icon	Meaning
⊞	Table
⤜	Foreign key
⊡	Object

To track data changes further, you can use the functions below.

- Show a specific object from the change history

  Select the entry for the object in the log and click 🔍. Loads the object and opens the overview form.

- Show a referenced object from the change history

  - Select the entry for the object in the log and click 🧩. The display switches to the originally referenced object and opens the overview form.

  - Select the entry for the object in the log and click 🧩. The display switches to the newly referenced object and opens the overview form.

Related topics

- Process information layout on page 44
- Analyzing data changes in reports and the TimeTrace on page 28

Schedules in One Identity Manager

Frequently, you need to run processes and calculation tasks at specified time intervals. To make this possible, you can define schedules in One Identity Manager. For example, schedules are required for scheduled running of processes within process handling or for different calculation tasks in One Identity Manager. A schedule can be in control of several tasks. Activation times are configured in a schedule for the tasks to be run.

You create and edit schedules in the Designer or in the Manager. The Designer displays all schedules of the system. You can edit individual schedules, such as schedules for attestation or schedules for compliance calculations, in the Manager. For detailed information about editing schedules in the Manager, refer to the administration guides for the modules.

Schedules are already defined in the default installation of One Identity Manager. Configure these according to your custom requirements.

NOTE: If you run a schedule, all tasks to which the schedule is assigned are run. Before you use a schedule on a repeated basis, check the effects of the process handling.

Related topics

Enabling and disabling schedules

You can edit individual schedules in the Manager. For detailed information about editing schedules in the Manager, refer to the administration guides for the modules.

Prerequisites for running schedules automatically

Enabled schedules are run automatically if the **QBM | Schedules** configuration parameter is set (default).

- In the Designer, check if the configuration parameter is set. If not, set the configuration parameter.

To enable a single schedule in the Designer

1. In the Designer, select the **Base data > General > Schedules** category.
2. Select the schedule.
3. Set **Enabled**.
4. Select the **Database > Save to database** and click **Save**.

To disable a single schedule in the Designer

5. In the Designer, select the **Base data > General > Schedules** category.
6. Select the schedule.
7. Disable the **Enable** option.
8. Select the **Database > Save to database** and click **Save**.

To temporarily stop schedules from running automatically

- In the Designer, disabled the **QBM | Schedules** configuration parameter.

  This stops the schedules from being run automatically. However, you can still start schedules manually.

Related topics

-

Starting schedules immediately

You can edit individual schedules in the Manager. For detailed information about editing schedules in the Manager, refer to the administration guides for the modules.

NOTE:

- Before you start a schedule manually, check whether other processes will be run as a result, that also need to be preprocessed by One Identity Manager.

- The last processing time is not updated when the schedule is started manually.

To start a schedule in the Designer immediately

1. In the Designer, select the **Base data > General > Schedules** category.

2. Select the schedule.

3. Click **Start**.

4. Confirm the security prompt with **Yes**.

Creating and editing schedules

You can edit individual schedules in the Manager. For detailed information about editing schedules in the Manager, refer to the administration guides for the modules.

To create a schedule in the Designer

1. In the Designer, select the **Base data > General > Schedules** category.

2. Select the **Object > New** menu item to create a new schedules.

3. Edit the schedule's main data.

4. Select the **Database > Save to database** and click **Save**.

To edit a schedule in the Designer

1. In the Designer, select the **Base data > General > Schedules** category.

2. Select a schedule.

3. Edit the schedule's main data.

4. Select the **Database > Save to database** and click **Save**.

Detailed information about this topic

- Properties of schedules on page 49

Properties of schedules

Enter the following properties for a schedule.

Table 16: Schedule properties

Property	Meaning
Name	Schedule ID. Translate the given text using the 🌐 button.
Description	Detailed description of the schedule. Translate the given text using the 🌐 button.

Property	Meaning
Table	Table whose data can be used by the schedule.
Enabled	Specifies whether the schedule is enabled. NOTE: Only active schedules are run. Active schedules are only run if the **QBM | Schedules** configuration parameter is set.
Time zones	Unique identifier for the time zone that is used for running the schedule. Choose between **Universal Time Code** or one of the time zones in the menu. NOTE: When you add a new schedule, the time zone is preset to that of the client from which you started the Designer.
Start (date)	The day on which the schedule should be run for the first time. If this day conflicts with the defined interval type, the first run is on the next available day based on the start date.
Validity period	Period within which the schedule is run. • If the schedule will be run for an unlimited period, select the **Unlimited duration** option. • To set a validity period, select the **Limited duration** option and enter the day the schedule will be run for the last time in **End (date)**.

Occurs	Interval in which the task is run. Other settings may be required depending on the settings.
	
	• **Every minute**: The schedule is run once a minute. The starting point is calculated from the rate of occurrence and the interval type.
	
	• **Hourly**: The schedule is run at defined intervals of a multiple of hours such as every two hours.
	    • Under **Repeat every**, specify after how many hours the schedule is run again.
	    • The starting point is calculated from the rate of occurrence and the interval type.
	
	• **Daily**: The schedule is run at specified times in a defined interval of days such as every second day at 6am and 6pm.
	    • Under **Start time**, specify the times to run the schedule.
	    • Under **Repeat every**, specify after how many days the schedule is run again.
	
	• **Weekly**: The schedule is run at a defined interval of weeks, on a specific day, at a specified time such as every second week on Monday at 6am and 6pm.
	    • Under **Start time**, specify the times to run the schedule.

- Under **Repeat every**, specify after how many weeks the schedule is run again.
- Specify the set day of the week for running the schedule.

- **Weekly**: The schedule is run at a defined interval of months, on a specific day, at a specified time such as every second month on the 1st and the 15th at 6am and 6pm.

    - Under **Start time**, specify the times to run the schedule.
    - Under **Repeat every**, specify after how many months the schedule is run again.
    - Specify the days of the month (1st - 31st of the month).

    NOTE: If the **Monthly** interval type with the sub interval **29**, **30** or **31** does not exist in this month, the last day of the month is used.

    Example:

    A schedule that is run on the 31st day of each month is run on April 30th. In February, the schedule is run on the 28th (or 29th in leap year).

- **Yearly**: The schedule is run at a defined interval of years, on a specific day, at a specified time such as every year on the 1st, the 100th, and the 200th day at 6am and 6pm.

    - Under **Start time**, specify the times to run the schedule.
    - Under **Repeat every**, specify after how many years the schedule is run again.
    - Specify the days of the year (1st - 366th day of the year).

        NOTE: If you select the 366th day of the year, the schedule is only run in leap years.

- **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday**, **Sunday**: The schedule is run on a defined day of the week, in specified months, at specified times such as every second Saturday in January and June at 10am.

    - Under **Start time**, specify the times to run the schedule.
    - Under **Repeat every**, specify after how many days of the month the schedule is run again. The values **1** to **4**, **-1** (last day of the week), and **-2** (last day but one of the week) are permitted.
    - Specify in which month to run the schedule. The values **1** to **12** are permitted. If the value is empty, the schedule is run each month.

Start time   Fixed start time Enter the time in local format for the chosen time zone. If

Property	Meaning
	there is a list of start times, the schedule is started at each of the given times.
Repeat every	Rate of occurrence for running the schedule within the selected time interval.
Last planned run/Next planned run	Activation time calculated by the DBQueue Processor. Activation times are recalculated whilst the schedule is running. The time of the next run is calculated from the interval type, rate of occurrence, and the start time. NOTE: One Identity Manager provides the start information in the time zone of the client where the program was started. Changes due to daylight saving are taken into account.

Related topics

- Enabling and disabling schedules on page 47

Calculating the activation time

The database schedule `QBM_PWatchDog` on `<database>` verifies the schedules that need to be run and their activation times, at regular intervals. When the database scheduler is run, all tasks are found that are within the valid time period and are enabled. A task is queued in the DBQueue for each schedule to be run. Then the time for the next scheduled run is calculated through the database schedule and entered in the schedule.

For tasks with the **Every minute** and **Hourly** interval types, the next planned activation time will be determined from the time at which the database schedule runs, how often it runs, and the specified time zone. For schedules with the interval types **Daily**, **Weekly**, **Monthly** and **Yearly**, the next planned activation time will be determined from the current day, the specified subinterval and the start time within the specified time zone.

Behavior of new schedules

When a new active schedule is added, the next scheduled run is calculated immediately. This is calculated on the basis on the start date or the current date of the next scheduled run. The time between runs is not taken into account.

The task is run if the activation time has been reached. When the next scheduled run is calculated, this time the interval will be taken into account.

Behavior of modified schedules

If a schedule changes, the next scheduled run is calculated immediately. This is calculated on the basis on the start date or the current date of the next scheduled run. The time between runs is not taken into account.

The task is run if the activation time has been reached. When the next scheduled run is calculated, this time the interval will be taken into account.

Scheduled maintenance tasks

Some calculation tasks for the DBQueue Processor are scheduled. There are schedules set up for these maintenance tasks, which you can customize as required. It is recommended to run maintenance task outside main working hours of the connected clients.

Table 17: DBQueue Processor maintenance tasks

Task	Schedule	Frequency
Reduce size of change entries	Reduce logs	Daily
Reduce size of process tracking logs	Reduce logs	Daily
Purge dynamic users	Reduce logs	Daily
Reduce size of process log entries	Reduce logs	Daily
Reduce size of process history	Reduce logs	Daily
Populate calendar	Daily maintenance tasks	Daily
Lock table statistics	Daily maintenance tasks	Daily
Calculate table statistics	Daily maintenance tasks	Daily
Rebuild table index NOTE: Reindexing of tables does not take place anymore if they are larger than 1 GB or have more than 1 million data records. Maintenance of these tables must be performed by the database administrator within maintenance periods.	Daily maintenance tasks	Daily
Delete closed cases in the IT Shop	Daily maintenance tasks	Daily
Clean up DBQueue Processor buffer	Daily maintenance tasks	Daily
Calculate statistics for data contents	Weekly maintenance	Weekly

Task	Schedule	Frequency
	tasks	
Set RowLock	Weekly maintenance tasks	Weekly

Related topics

- Schedules in One Identity Manager on page 47

Mail templates in One Identity Manager

One Identity Manager provides the means to send email notifications. For example, notifications can be sent from process handling, about attestation or the status of IT Shop requests.

You use mail templates to design the appearance and content of email notifications. A mail template consists of general main data such as target format, important, or mail notification confidentiality and one or more mail definitions. The mail text is defined in several languages in the mail template. The recipient's language preferences are taken into account when an email notification is generated.

Create and edit mail templates in the Designer or in the Manager. The Designer displays all mail templates of the system. You can edit individual mail templates such as mail templates for requests in IT Shop or mail templates for attestation in the Manager. For detailed information about editing mail templates in the Manager, refer to the administration guides for the modules.

A Designer is integrated in the Manager and in the Mail Template Editor to simplify writing notifications. In the Mail Template Editor you can create email texts with Microsoft Word style editing and formatting functions and a preview of the email.

Email notifications are generated through default processes during process handling. To use email notifications based on mail templates for other business procedures, for example creating user accounts, you have to create custom mail templates and processes. Use the `MailComponent` process component to provide the `SendRichMail` process task for this purpose.

Related topics

Creating and editing mail templates

You can edit certain mail templates in the Manager. For detailed information about editing mail templates in the Manager, refer to the administration guides for the modules.

To edit a mail template in the Designer

1. In the Designer, select the **Mail templates** category.
2. Select the mail template and start Mail Template Editor using the **Edit mail template** task.

To create a new mail template in the Designer

1. In the Designer, select the **Mail templates** category.
2. Start Mail Template Editor using the **Create a new mail template** task.

Related topics

- Copying mail templates on page 56
- Creating mail previews on page 57

Copying mail templates

You can edit certain mail templates in the Manager. For detailed information about editing mail templates in the Manager, refer to the administration guides for the modules.

To copy a mail template in the Designer

1. In the Designer, select the **Mail templates** category.
2. Select the mail template you want to copy and start the Mail Template Editor using the **Edit mail template**.
3. Select the **Mail template > Copy mail template** menu item.
4. Enter the name of the new mail template and click **OK**.

   The new mail template is displayed in the Mail Template Editor. Now, you can edit the mail template.

Related topics

- Creating and editing mail templates on page 56
- Creating mail previews on page 57

Creating mail previews

You can edit certain mail templates in the Manager. For detailed information about editing mail templates in the Manager, refer to the administration guides for the modules.

To display a mail template preview in the Designer

1. In the Designer, select the **Mail templates** category.

2. Select the mail template and start Mail Template Editor using the **Edit mail template** task.

3. Select the **Mail templates > Mail preview** menu item.

4. Select the base object and click **OK**.

General properties of mail templates

Table 18: Mail template properties

Property	Meaning
Mail template	Name of the mail template. This name will be used to display the mail templates in the administration tools and in the Web Portal. Translate the given text using the 🌐 button.
Base object	Mail template base object. A base object only needs to be entered if the mail definition properties of the base object are referenced.
Report (parameter set)	Report, made available through the mail template.
Description	Mail template description. Translate the given text using the 🌐 button.
Target format	Format in which to generate email notification. Permitted values are: • **HTML**: The email notification is formatted in HTML. Text formats, for example, different fonts, colored fonts, or other text formatting, can be included in HTML format. • **TXT**: The email notification is formatted as text. Text format does not support bold, italics, or colored font, or other text formatting. Images displayed directly in the message are not supported.
Design type	Design in which to generate the email notification. Permitted values are: • **Mail template**: The generated email notification contains the mail body in accordance with the mail definition. • **Report**: The generated email notification contains the report

Property	Meaning
	specified under **Report (parameter set)** as its mail body.
	• **Mail template, report in attachment**: The generated email notification contains the mail body in accordance with the mail definition. The report specified under **Report (parameter set)** is attached to the notification as a PDF file.
Importance	Importance for the email notification. Permitted values are **Low**, **Normal**, and **High**.
Confidentiality	Confidentiality for the email notification. Permitted values are **Normal**, **Personal**, **Private**, and **Confidential**.
Can unsubscribe	Specifies whether the recipient can unsubscribe email notification. If this option is set, the emails can be unsubscribed through the Web Portal.
Deactivated	Specifies whether this mail template is disabled.
Mail definition	Selects the mail definition in a specific language. NOTE: If the **Common | MailNotification | DefaultCulture** configuration parameter is set, the mail definition is loaded in the default language for email notifications when the template is opened.
Language	Language that applies to the mail template. The recipient's language preferences are taken into account when an email notification is generated.
Subject	Subject of the email message.
Mail body	Content of the email message.

Related topics

- Creating and editing mail definitions on page 58

Creating and editing mail definitions

Mail texts can be defined in these different languages in a mail template. This ensures that the language of the recipient is taken into account when the email is generated.

To create a new mail definition

1. In the Designer, select the **Mail templates** category.

2. Select the mail template and start Mail Template Editor using the **Edit mail template** task.

3. In the result list, select the language for the mail definition in the **Language** menu.

   All active languages are shown. To use another language, in the Designer, enable the corresponding countries. For more information, see the *One Identity Manager Configuration Guide*.

4. Enter the subject in **Subject**.

5. Edit the mail text in the **Mail definition** view with the help of the Mail Text Editor.

6. Save the changes.

To edit an existing mail definition

1. In the Designer, select the **Mail templates** category.

2. Select the mail template and start Mail Template Editor using the **Edit mail template** task.

1. In the **Mail definition** menu, select the language for the mail definition.

   NOTE: If the **Common | MailNotification | DefaultCulture** configuration parameter is set, the mail definition is loaded in the default language for email notifications when the template is opened.

2. Edit the mail subject line and the body text.

3. Save the changes.

Related topics

- Defining default fonts and default font sizes for mail templates on page 66
- Creating and editing mail templates on page 56
- Using base object properties in mail templates on page 59
- Using hyperlinks to the Web Portal in mail templates on page 61
- Default functions for creating hyperlinks on page 61
- Using scripts in mail templates on page 64
- Support for dynamically generated HTML code in mail templates on page 64
- Usingprocess parameters in hyperlinks on page 65
- Email signature configuration for mail templates on page 66

Using base object properties in mail templates

In the subject line and body text of a mail definition, you can use all properties of the object entered under **Base object**. You can also use the object properties that are referenced by foreign key relation.

To access properties use dollar notation. For more information, see the *One Identity Manager Configuration Guide*.

Example:

An IT Shop requester should receive email notification about the status of the request.

Table 19: Email notification properties

Property	Value
Base object	PersonWantsOrg
Subject	"$DisplayOrg[D]$" status change
Mail body	Dear $FK(UID_PersonOrdered).Salutation[D]$ $FK(UID_PersonOrdered).FirstName$ $FK(UID_PersonOrdered).LastName$,
	The status was changed on the following request on $DateHead:Date$.
	Requested by: $DisplayPersonInserted$
	Requested by: $DisplayPersonInserted$
	Reason: $OrderReason$
	Current status of your request:
	Approval: granted
	Approver: $DisplayPersonHead[D]$
	Reason: $ReasonHead[D]$

The generated email notification could look like the following, for example, once it has been formatted.

Subject: "Service Notebook" status change

Dear Ms Monica Flaster,

The status was changed on the following request on 03/08/2011 11:14:53.

Product: Service Notebook
Requested by: Fletcher, Monica
Reason: For on-site processing

Current status for your request:

Approval: granted
Approver: Rippington, Rudiger
Reason: approved

Related topics

- Creating and editing mail definitions on page 58

Using hyperlinks to the Web Portal in mail templates

You can add hyperlinks to the Web Portal in the mail text of a mail definition. If the recipient clicks on the hyperlink in the email, the Web Portal opens on that web page and further actions can be carried out. In the default version, this method is implemented for IT Shop requests, in Identity Audit, policy checks and attestations.

Prerequisites for using this method

- The **QER | WebPortal | BaseURL** configuration parameter is enabled and contains the URL path to the Web Portal. You edit the configuration parameter in the Designer.

  http://<server name>/<application>

  with:

  <server name> = name of server

  <application> = path to the Web Portal installation directory

To add a hyperlink to the Web Portal in the mail text

1. Click the position in the mail text of the mail definition where you want to insert a hyperlink.

2. Open the **Hyperlink** context menu and enter the following information.

   - **Display text**: Enter a caption for the hyperlink.
   - **Link to**: Select the **File or website** option.
   - **Address**: Enter the address of the page in the Web Portal that you want to open.

     NOTE: One Identity Manager provides a number of default functions that you can use to create hyperlinks in the Web Portal.

3. To accept the input, click **OK**.

Related topics

- Creating and editing mail definitions on page 58
- Default functions for creating hyperlinks on page 61
- Usingprocess parameters in hyperlinks on page 65

Default functions for creating hyperlinks

Several default functions are available to help you create hyperlinks. You can use the functions directly when you add a hyperlink in the mail body of a mail definition or in processes

Direct function input

You can reference a function when you add a hyperlink in the **Address** field of the **Hyperlink** context menu.

$Script(<Function>)$

Example:

$Script(VI_BuildITShopLink_Show_for_Requester)$

$Script(VI_BuildAttestationLink_Approve)$

$Script(VI_BuildComplianceLink_Show)$

$Script(VI_BuildQERPolicyLink_Show)$

Default functions for requests

The `VI_BuildAttestationLinks` script contains a collection of default functions for composing hyperlinks to directly grant or deny approval of requests from email notifications.

Table 20: Functions of the VI_BuildAttestationLinks script

Function	Usage
VI_BuildAttestationLink_Show	Opens the attestation page in the Web Portal.
VI_BuildAttestationLink_Approve	Approves an attestation and opens the attestation page in the Web Portal.
VI_BuildAttestationLink_Deny	Denies an attestation and opens the attestation page in the Web Portal.
VI_BuildAttestationLink_AnswerQuestion	Opens the page for answering a question in the Web Portal.
VI_BuildAttestationLink_Pending	Opens the page with pending attestations in the Web Portal.

Default functions for IT Shop requests

The `VI_BuildITShopLinks` script contains a collection of default functions for composing hyperlinks to directly grant or deny approval of IT Shop requests from email notifications.

Table 21: Functions of the VI_BuildITShopLinks script

Function	Usage
VI_BuildITShopLink_Show_for_Approver	Opens the overview page for request approval in the Web Portal.
VI_BuildITShopLink_Show_for_Requester	Opens the overview page for requests in the Web Portal.
VI_BuildITShopLink_	Approves a request and opens the approvals page in the Web

Function	Usage
Approve	Portal.
VI_BuildITShopLink_ Deny	Denies a request and opens the approvals page in the Web Portal.
VI_BuildITShopLink_ AnswerQuestion	Opens the page for answering a question in the Web Portal.
VI_BuildITShopLink_ Reject	Opens the page with denied requests in the Web Portal.
VI_ BuildAttestationLink_ Pending	Opens the page with pending requests in the Web Portal.
VI_BuildITShopLink_ Unsubscribe	Creates the link for canceling email notification. This function is used in processes for unsubscribing email notifications.

Default functions for identity audit

The VI_BuildComplianceLinks script contains a collection of default functions for composing hyperlinks for exception approval of rule violations.

Table 22: Functions of the VI_BuildComplianceLinks script

Function	Usage
VI_BuildComplianceLink_Show	Opens the exception approval page in the Web Portal.

Default function for policy checking

The VI_BuildComplianceLinks script contains a collection of default functions for composing hyperlinks for exception approval of policy violations.

Table 23: Functions of the VI_BuildComplianceLinks script

Function	Usage
VI_BuildQERPolicyLink_Show	Opens the exception approval page in the Web Portal.

Related topics

Using scripts in mail templates

For more information about using scripts, see the *One Identity Manager Configuration Guide*.

In mail templates, any parameters can be used when calling a script.

Syntax

$SCRIPT(ScriptName, "Options")$

The Options parameter is optional and is passed as a string. Custom parameters can be coded in any way in this string. Quotes ("") are masked by doubling. In the script, the parameter is passed as the second parameter after the base object. The base object can be either IEntity or ISingleDbObject.

Example:

Public Function CCC_Script(baseEntity as IEntity, options as String) as String

Dim arr = options.Split("|"c)

Dim p1 = arr(0)

Dim p2 = arr(1)

End Function

Using the script in mail templates.

$SCRIPT(CCC_Script, "Param1|Param2")$

Support for dynamically generated HTML code in mail templates

For detailed information about using dollar ($) notation, see the *One Identity Manager Configuration Guide*.

In dollar notation, you can select the **HTML** type. The HTML code is accepted in scripts and columns but not masked. There is no security check.

> **Example: Script with HTML code**
>
> ```
> Public Function CCC_HtmlMailText(obj As IEntity) As String
> Return "<h1 style='color:red'>" & obj.Display & "</h1>"
> End Function
> ```
>
> **Using the script in mail templates.**
>
> ```
> $SCRIPT(CCC_HtmlMailText):HTML$
> ```

Usingprocess parameters in hyperlinks

Use this method to pass additional parameters to a function. Email notifications are generated during the process handling. Use the `MailComponent` process component to provide the `SendRichMail` process task for this purpose.

To compile a hyperlink in a process, for example, cancellation of email notifications, use the `[ParamName 1-n]` and `[ParamValue 1-n]` free process parameters of the process component.

NOTE: By default, 10 pairs of parameters are available. If this number is not sufficient, you can create additional custom process parameters, which you can then use as parameters in the Process Editor.

> **Example: Populating process parameters**
>
> ParamName1: Value = "NoSubscription"
>
> ParamValue1: Value = VI_BuildITShopLink_Unsubscribe (values("UID_RichMail").ToString())
>
> UID_RichMail is determined by the pre-script for generating within the process and passed to the function.
>
> Take implementation examples from base object `PersonWantsOrg` processes that are triggered by changes to IT Shop requests.

The process parameter is referenced when a hyperlink is inserted in a mail definition using the **Hyperlink** menu in the **Address input** field:

```
$PC(<ParamName>)$
```

> **Example:**
>
> $PC(NoSubscription)$

For more information about creating and editing processes, see the *One Identity Manager Configuration Guide*.

Related topics

Defining default fonts and default font sizes for mail templates

To define default fonts for mail templates

- In the Designer, set the **Common | MailNotification | DefaultFont** configuration parameter and enter a font. The default value is **Time New Roman**.

To define default font sizes for mail templates

- In the Designer, set the **Common | MailNotification | DefaultFontSize** configuration parameter and enter a font size. The default value is **12**.

Email signature configuration for mail templates

Configure the email signature for mail templates using the following configuration parameters. Edit the configuration parameters in the Designer.

Table 24: Configuration parameters for email signatures

Configuration parameter	Description
Common	MailNotification	Signature	Data for the signature in email automatically generated from mail templates.
Common	MailNotification		Signature under the salutation.

Configuration parameter	Description
Signature | Caption	
Common | MailNotification | Signature | Company	Company name.
Common | MailNotification | Signature | Link	Link to the company's website.
Common | MailNotification | Signature | LinkDisplay	Display text for the link to the company's website.

`VI_GetRichMailSignature` combines the components of an email signature according to the configuration parameters for use in mail templates.

Password policies in One Identity Manager

One Identity Manager provides you with support for creating complex password policies, for example, for system user passwords, the employees' central password as well as passwords for individual target systems. Password polices apply not only when the user enters a password but also when random passwords are generated.

Predefined password policies are supplied with the default installation that you can use or customize if required. You can also define your own password policies.

Create and edit mail password policies in the Designer or in the Manager. The Designer displays all password policies of the system. You can edit individual password policies, such as password policies for target systems or password policies for the central password of employees, in the Manager.

For detailed information about password policies for employees, see the *One Identity Manager Identity Management Base Module Administration Guide*. For detailed information about password policies for user accounts, see the administration guides of the target systems.

Detailed information about this topic

Predefined password policies

You can customize predefined password policies to meet your own requirements if necessary.

Password for logging in to One Identity Manager

The **One Identity Manager password policy** is applied for logging in to One Identity Manager. This password policy defines the settings for the system user passwords (`DialogUser.Password` and `Person.DialogUserPassword`) as well as the passcode for a one time log in on the Web Portal (`Person.Passcode`).

NOTE: The **One Identity Manager password policy** is marked as the default policy. This password policy is applied if no other password policy can be found for employees, user accounts, or system users.

For detailed information about password policies for employees, see the *One Identity Manager Identity Management Base Module Administration Guide*.

Password policy for forming employees' central passwords

An employee's central password is formed from the target system specific user accounts by respective configuration. The **Employee central password policy** defines the settings for the (`Person.CentralPassword`) central password. Members of the **Identity Management | Employees | Administrators** application role can adjust this password policy.

IMPORTANT: Ensure that the **Employee central password policy** does not violate the target system-specific requirements for passwords.

For detailed information about password policies for employees, see the *One Identity Manager Identity Management Base Module Administration Guide*.

Password policies for user accounts

Predefined password policies are provided, which you can apply to the user account password columns of the user accounts. You can define password policies for user accounts for various base objects, for example, for account definitions, manage levels, or target systems.

For detailed information about password policies for user accounts, see the administration guides of the target systems.

Using password policies

You can assign password policies to system user passwords, the employees' central password as well as passwords for individual target systems. Assign a password policy to the base object to which it should apply.

- The predefined **One Identity Manager password policy** password policy is assigned to the (DialogUser.Password and Person.DialogUserPassword) system user passwords as well as the passcode of the employee (Person.Passcode).

- The predefined password policy **Employee central password policy** is assigned to the employee's central password (Person.CentralPassword).

- The password policies for target systems are assigned to the password columns of the user accounts.

For more information about password policies for employees, see the *One Identity Manager Identity Management Base Module Administration Guide*. For detailed information about password policies for user accounts, see the administration guides of the target systems.

> NOTE:
>
> - In the QBMVPwdPolicyColumns view, you define which base objects and password columns are permitted for password policies and the order in which the password policies are to be applied. If necessary, you can add your own references to customize the view in the Designer.
>
> - If you create new custom tables with password columns, in the Designer, assign the VI.Common.Customizer.PwdPolicyColumnEntityLogic customizer to the table definition.

For more information, see the *One Identity Manager Configuration Guide*.

If you want to apply another password policy to the password columns, change the password policy assignment to the base object.

To change a password policy's assignment

1. In the Designer, select the **Base data > Security settings > Password policies** category.

2. Select the password policy in the result list.

3. Select the **Assign objects** task.

4. In the **Assignments** pane, select the assignment you want to change.

5. From the **Password Policies** menu, select the new password policy you want to apply.

6. Save the changes.

To reassign a password policy

1. In the Designer, select the **Base data > Security settings > Password policies** category.

2. Select the password policy in the result list.

3. In the **Assignments** pane, click **Add** and enter the following data.

Table 25: Assigning a password policy

Property	Description
Password column	The password column's identifier.

Apply to	Application scope of the password policy.
	
	To specify an application scope
	
	  1. Click the **...** button beside the input field.
	
	  2. Select the table which contains the password column under **Table**.
	
	  3. Select the specific base objects under **Apply to**.
	
	  4. Click **OK**.

4. Save the changes.

Editing password policies

Predefined password policies are supplied with the default installation that you can use or customize if required.

To edit a password policy

1. In the Designer, select the **Base data > Security settings > Password policies** category.

2. In the List Editor, select the password policy.

3. Edit the password policy's main data.

4. Save the changes.

Detailed information about this topic

- General main data of password policies on page 72
- Policy settings on page 73
- Character classes for passwords on page 74
- Custom scripts for password requirements on page 76

Creating password policies

Predefined password policies are supplied with the default installation that you can use or customize if required. You can also define your own password policies.

To create a password policy

1. In the Designer, select the **Base data > Security settings > Password policies** category.

2. Select the **Object > New** menu item to create a new password policy.

3. On the main data form, enter the main data of the password policy.

4. Save the changes.

Detailed information about this topic

- General main data of password policies on page 72
- Policy settings on page 73
- Character classes for passwords on page 74
- Custom scripts for password requirements on page 76

General main data of password policies

Enter the following main data of a password policy.

Table 26: main data for a password policy

Property	Meaning
Display name	Password policy name. Translate the given text using the 🌐 button.
Description	Text field for additional explanation. Translate the given text using the 🌐 button.
Error Message	Custom error message generated if the policy is not fulfilled. Translate the given text using the 🌐 button.
Owner (Application Role)	Application roles whose members can configure the password policies.
Default policy	Mark as default policy for passwords. This option cannot be changed. NOTE: The **One Identity Manager password policy** is marked as the default policy. This password policy is

Property	Meaning
	applied if no other password policy can be found for employees, user accounts, or system users.

Related topics

- Creating password policies on page 72

Policy settings

Define the following settings for a password policy on the **Password** tab.

Table 27: Policy settings

Property	Meaning
Initial password	Initial password for newly created user accounts. The initial password is used if a password is not entered when you create a user account or if a random password is not generated.
Password confirmation	Reconfirm password.
Minimum Length	Minimum length of the password. Specify the number of characters a password must have. If the value is **0**, no password is required.
Max. length	Maximum length of the password. Specify the number of characters a password can have. The maximum permitted value is **256**.
Max. errors	Maximum number of errors. Set the number of invalid passwords attempts. The number of failed logins is only taken into account when logging in to One Identity Manager. If the value is **0**, the number of failed logins is not taken into account.
	This data is only taken into account if the One Identity Manager login was through a system user or employee based authentication module. If a user has exceeded the maximum number of failed logins, the employee or system user will not be able to log in to One Identity Manager.
	You can use the Password Reset Portal to reset the passwords of employees and system users who have been blocked. For more information, see the *One Identity Manager Web Designer Web Portal User Guide*.

Property	Meaning
Validity period	Maximum age of the password. Enter the length of time a password can be used before it expires. If the value is **0**, then the password does not expire.
Password history	Enter the number of passwords to be saved. If, for example, a value of **5** is entered, the user's last five passwords are stored. If the value is **0**, then no passwords are stored in the password history.
Minimum password strength	Specifies how secure the password must be. The higher the password strength, the more secure it is. The value **0** means that the password strength is not tested. The values **1**, **2**, **3** and **4** specify the required complexity of the password. The value **1** represents the lowest requirements in terms of password strength. The value **4** requires the highest level of complexity.
Name properties denied	Specifies whether name properties are permitted in the password. If this option is set, name properties are not permitted in passwords. The values of these columns are taken into account if the **Contains name properties for password check** option is set. In the Designer, adjust this option in the column definition. For more information, see the *One Identity Manager Configuration Guide*.

Related topics

- Creating password policies on page 72

Character classes for passwords

Use the **Character classes** tab to specify which characters are permitted for a password.

Table 28: Character classes for passwords

Property	Meaning
Required number of character classes	Number of rules for character classes that must be fulfilled so that a password adheres to the password policy. The following rules are taken into account for **Min. number letters**, **Min. number lowercase**, **Min. number uppercase**, **Min. number digits**, and **Min. number special characters**. That means: - Value **0**: All character class rules must be fulfilled.

Property	Meaning
	• Value **>0**: Minimum number of character class rules that must be fulfilled. At most, the value can be the number of rules with a value **>0**. NOTE: Generated passwords are not tested for this.
Min. number letters	Specifies the minimum number of alphabetical characters the password must contain.
Min. number lowercase	Specifies the minimum number of lowercase letters the password must contain.
Min. number uppercase	Specifies the minimum number of uppercase letters the password must contain.
Min. number digits	Specifies the minimum number of digits the password must contain.
Min. number special characters	Specifies the minimum number of special characters the password must contain.
Permitted special characters	List of permitted special characters.
Max. identical characters in total	Specifies the maximum number of identical characters that can be present in the password in total.
Max. identical characters in succession	Specifies the maximum number of identical character that can be repeated after each other.
Denied special characters	List of special characters that are not permitted.
Do not generate lowercase letters	Specifies whether a generated password can contain lowercase letters. This setting only applies when passwords are generated.

Property	Meaning
Do not generate uppercase letters	Specifies whether a generated password can contain uppercase letters. This setting only applies when passwords are generated.
Do not generate digits	Specifies whether a generated password can contain digits. This setting only applies when passwords are generated.
Do not generate special characters	Specifies whether a generated password can contain special characters. If this option is set, only letters, numbers, and spaces are allowed in passwords. This setting only applies when passwords are generated.

Related topics

Custom scripts for password requirements

You can implement custom scripts for testing and generating passwords if the password requirements cannot be mapped with the existing settings options. Scripts are applied in addition to the other settings.

Detailed information about this topic

Checking passwords with a script

You can implement a script if additional policies need to be used for checking a password that cannot be mapped with the available settings.

Syntax of check scripts

Public Sub CCC_CustomPwdValidate(policy As VI.DB.Passwords.PasswordPolicy, spwd As System.Security.SecureString)

With parameters:

policy = password policy object

spwd = password to check

TIP: To use a base object, take the `Entity` property of the `PasswordPolicy` class.

Example: Script that checks a password

A password cannot start with **?** or **!** . The password cannot start with three identical characters. The script checks a given password for validity.

```
Public Sub CCC_PwdValidate( policy As VI.DB.Passwords.PasswordPolicy, spwd As
System.Security.SecureString)

    Dim pwd = spwd.ToInsecureArray()

    If pwd.Length>0

        If pwd(0)="?" Or pwd(0)="!"

            Throw New Exception(#LD("Password can't start with '?' or
            '!'")#)

        End If

    End If

    If pwd.Length>2

        If pwd(0) = pwd(1) AndAlso pwd(1) = pwd(2)

            Throw New Exception(#LD("Invalid character sequence in
            password")#)

        End If

    End If

End Sub
```

To use a custom script for checking a password

1. In the Designer, create your script in the **Script Library** category.
2. Edit the password policy.
   a. In the Designer, select the **Base data > Security settings > Password policies** category.
   b. In the List Editor, select the password policy.
   c. On the **Scripts** tab, enter the name of the script to be used to check a password in the **Check script** field.
   d. Save the changes.

Related topics

- Generating passwords with a script on page 78
- Creating password policies on page 72

Generating passwords with a script

You can implement a generating script if additional policies need to be used for generating a random password, which cannot be mapped with the available settings.

Syntax for generating script

Public Sub CCC_PwdGenerate(policy As VI.DB.Passwords.PasswordPolicy, spwd As System.Security.SecureString)

With parameters:

policy = password policy object

spwd = generated password

TIP: To use a base object, take the `Entity` property of the `PasswordPolicy` class.

Example: Script that generates a password

The script replaces invalid **?** and **!** characters at the beginning of random passwords with **_**.

```
Public Sub CCC_PwdGenerate( policy As VI.DB.Passwords.PasswordPolicy, spwd As System.Security.SecureString)

    Dim pwd = spwd.ToInsecureArray()

    ' replace invalid characters at first position

    If pwd.Length>0

        If pwd(0)="?" Or pwd(0)="!"

            spwd.SetAt(0, CChar("_"))

        End If

    End If

End Sub
```

To use a custom script for generating a password

1. In the Designer, create your script in the **Script Library** category.

2. Edit the password policy.

   a. In the Designer, select the **Base data > Security settings > Password policies** category.

   b. In the List Editor, select the password policy.

   c. On the **Scripts** tab, enter the name of the script to be used to generate a

password in the **Generating script** field.

d. Save the changes.

Related topics

- Checking passwords with a script on page 76
- Creating password policies on page 72

Editing the excluded list for passwords

You can add words to a list of restricted terms to prohibit them from being used in passwords.

| NOTE: The restricted list applies globally to all password policies.

To add a term to the restricted list

1. In the Designer, select the **Base data > Security settings > Password policies** category.

2. Create a new entry with the **Object > New** menu item and enter the term you want to exclude from the list.

3. Save the changes.

Checking passwords

When you verify a password, all the password policy settings, custom scripts, and the restricted passwords are taken into account.

To verify if a password conforms to the password policy

1. In the Designer, select the **Base data > Security settings > Password policies** category.

2. Select the password policy in the List Editor.

3. Select the **Test** tab.

4. Select the table and object to be tested in **Base object for test**.

5. Enter a password in **Enter password to test**.

   A display next to the password shows whether it is valid or not.

Testing the generation of passwords

When you generate a password, all the password policy settings, custom scripts and the restricted passwords are taken into account.

To generate a password that conforms to the password policy

1. In the Designer, select the **Base data > Security settings > Password policies** category.

2. In the List Editor, select the password policy.

3. Select the **Test** tab.

4. Click **Generate**.

   This generates and displays a password.

Password expiry

Employee and system user based authentication modules support password expiry. The columns Person.PasswordLastSet and DialogUser.PasswordLastSet contain the time and date that the password was last changed.

There are different ways to inform employees that their password is going to expire:

- Users are alerted about their password expiring when they log in to One Identity Manager and can change their password if necessary.

- For employee-based authentication modules, the system sends reminder notifications in relation to expiring passwords as of seven days in advance of the password expiry date.

  - You can adjust the time in days in the **Common | Authentication | DialogUserPasswordReminder** configuration parameter. Edit the configuration parameter in the Designer.

  - The notifications are triggered in accordance with the **Reminder system user password expires** schedule and use the **Employee - system user password expires** mail template. You can adjust the schedule and mail template in the Designer if required.

TIP: To prevent passwords expiring for service account, for example, you can set **Password never expires** (DialogUser.PasswordNeverExpires) in the Designer for the affected system users.

For detailed information about the One Identity Manager authentication modules and about editing system users, see the *One Identity Manager Authorization and Authentication Guide*.

Related topics

- Schedules in One Identity Manager on page 47
- Mail templates in One Identity Manager on page 55

Displaying locked employees and system users

If a user has exceeded the maximum number of failed logins, the employee or system user will not be able to log in to One Identity Manager.

- Locked employees are displayed in the Manager in the **Employees > Locked employees** category. An additional message referring to the locked login is also displayed on the overview form for an employee.

- Locked system users are displayed in the Designer in the **Permissions > System users > Locked system users** category. An additional message referring to the locked login is also displayed on the overview form for a system user.

You can reset the passwords of employees and system users who have been blocked in Password Reset Portal. This unlocks the employees and system users again. For more information, see the *One Identity Manager Web Designer Web Portal User Guide* and the *One Identity Manager Web Application Configuration Guide*.

Working with change labels

Define change labels under which changes are grouped together in order to swap data between development and test databases as well as the productive database. In the Database Transporter program, change labels are provided as an export criterion for creating custom configuration packages.

You can select individual objects from any objects in the database and book them to a change label. In certain cases, it is necessary to add the dependent objects to the change label as well. For example, if processes are being transported, the dependent process steps, process parameters, and events should also be transported. This is also true for approval policies, approval workflows, approval steps, and approval procedures.

Detailed information about this topic

Snapshots of objects or object references

Change labels contain changes to individual properties of an object at a certain point in time. Change labels may consist of:

- Snapshot of an object at a certain point in time, optionally with dependent objects (default)

  When you create a custom configuration package, the object is added to the transport package with all its properties. The properties contain the values given at the time they were added.

- Object reference, optionally with all its dependent objects

  When you create a custom configuration package, the object is added to the transport package with all its properties. The properties are determined at the time of export.

You specify whether an object is added to a change label as a snapshot or a reference when you save the assignment.

NOTE: Snapshots of objects and object references cannot be grouped together in one change label. Use different change labels for this.

Related topics

- Tips for working with change labels on page 83
- Assigning objects and dependencies to change labels on page 86
- Editing change labels on page 87

Tips for working with change labels

You can create and edit change labels in different One Identity Manager tools. Note the following tips for working with change labels.

- Be consistent about booking all the changes made to one object to the change label. It is not possible to add changes of individual properties to the change label at a later date.
- Snapshots of objects and object references cannot be grouped together in one change label. Use different change labels for this.
- Whether objects are assigned to a change label as snapshots or as references depends on the selected change labels
- Normally, objects can be assigned to a change label beforehand or on saving. For example, in the Designer, you can assign a change label when you commit changes

to the database.

- In the Manager and in the Designer, you can assign objects to change labels in their properties dialog.
- The Manager and in the Designer have their own edit dialog that allows more comprehensive editing of change labels.
- To use change labels in the Manager, the Manager must be running in expert mode.

Related topics

Creating and editing change labels

To create a change label

1. In the Manager or the Designer, select the **Database > Change management** menu item.
2. In the **Change management** dialog next to the **Change labels** menu, click ⬆.
3. In the **Change labels** dialog, create a new change label by clicking ⬛.
4. Enter the following label data.

Table 29: Change label properties

Property	Meaning
Change label	Change label name. This name is used to select the change label for allocating the changes or creating a customer transport package.
Description	Detailed description of the change label.
Parent change label	Specifies a parent label (optional).
State	Status of object changes, such as **Development**, **Test**, **Production**.
Status comments	Additional comments in relation to the status
Comment	Additional information to enable tracking of changes to a change label

Property	Meaning
Label type	Label type for more detailed classification Permitted values are **Change**, **Other**, **IT Shop**, **Keyword**, and **Release**. The **Change** label type is used by default.
Suspended	Specifies whether the change label is locked or closed. If a change label is locked, no further changes can be booked to this label. For example, change labels can be locked and closed after they have been transported with the Database Transporter.
Sort order	Order in which the data is presented. The sort order is used in the Database Transporter for importing data using change labels.

5. Click the 🖫 button.

6. Click **OK**.

   The **Change label** dialog closes. The change label is pre-selected in the **Change management** dialog in the **Change label** menu.

To edit a change label

1. In the Manager or the Designer, select the **Database > Change management** menu item.

2. In the **Change management** dialog next to the **Change labels** menu, click 🔼.

3. In the **Change management** dialog, select the change label and open the edit view with the 🖉 button.

4. Edit the change label data.

5. Click the 🖫 button.

6. Click **OK**.

   The **Change label** dialog closes. The change label is pre-selected in the **Change management** dialog in the **Change label** menu.

Related topics

- Assigning objects and dependencies to change labels on page 86
- Assigning change labels when saving in the Designer on page 88
- Editing change labels on page 87
- Displaying content of change labels on page 89
- Deleting change labels on page 90
- Release management on page 91

Assigning objects and dependencies to change labels

You can select individual objects from any objects in the database and book them to a change label. For example, use the Designer to book processes and their dependents to change labels and approval procedures. In the Manager, you can assign approval policies, approval workflows, approval steps, and approval procedures to change labels.

To assign objects and dependencies to a change label

1. In the Manager or the Designer, select the object and open the **Properties** context menu.

2. You can see which change labels the object belongs to on the **Change labels** tag.

   - To create a new change label assign it, select the **New change label** and enter a name for the change label in the input field.

   - To assign an existing change label, select the **Existing change label** option and select a change label from the menu.

3. (Optional) To assign dependent objects, click **Add dependent objects**.

   This opens a **Add dependent objects** dialog in that lists all the dependencies of the selected object.

   a. Click **Add dependent objects**.

   b. Check that all the required dependencies are enable.

   c. Enable other dependencies if required.

   > TIP: Use the ▤ button to switch between the relation and the selected objects and to identify the dependencies.

   d. Click **OK**.

4. Save the change label assignments. Use the arrow menu next to the **Save** menu to select the method.

   - **Assign object data**: (Default) A snapshot of the object is made at a specific point, with or without dependencies, and assigned to the change label.

   - **Assign object reference**: A reference to an object is created, with or without dependencies, and assigned to the change label.

5. Click **OK**.

Related topics

- Snapshots of objects or object references on page 83
- Tips for working with change labels on page 83
- Editing change labels on page 87

Editing change labels

In the Manager and the Designer, you can edit change label assignments. You can add objects and dependencies to change labels and remove them again.

To assign objects and dependencies to a change label

1. In the Manager or the Designer, select the **Database > Change management** menu item.

2. Select the change label in the **Change labels** menu in the **Change management** dialog.

3. In the **Table** list, select the database table from which you want to copy objects to the change label.

4. To limit the number of objects found

   a. Next to the **Table** menu, click the button ▼.

   b. Enter a condition in **Filter**.

      Enter the condition as a WHERE clause for a database query. You can enter the database query directly as in SQL or use the wizard, which you open by clicking on the 🔿 button next to the field.

   c. Click **Apply**.

5. To map dependent objects

   a. Next to the **Table** menu, click the button 🔿.

      This opens a separate selection window that displays the ChildRelation (CR), ForeignKey (FK) and many-to-many relations for the selected database table.

   b. Select the relevant table relations in **Table relations**.

      The objects that are connected by means of these table relations are also marked with the change label when an object is selected and assigned.

6. In the **Objects** pane, select your objects and assign them to the change label. In this case, the method depends on the change labels you selected.

   - If the change label you select is for snapshots of objects, click 🔿. (Default)

   - If the change label you selected already contains object references, click 🔿.

   | TIP: To select more than one object, use **Shift + select** or **Ctrl + select**.

Related topics

Removing objects from change labels

To remove objects from a change label

1. In the Designer or the Manager, select the **Database > Change management** menu item.

2. In the **Change management** dialog, use the **Change label** menu to select the change label.

3. Select the object that you want to remove from the change label.

   - If you selected a change label used for snapshots of objects, select the objects in the **Tagged changes** pane and click .

   - If you selected a change label used for object references, select the objects in the **Assigned objects** pane and click

   TIP: To select more than one object, use **Shift + select** or **Ctrl + select**.

Related topics

- Snapshots of objects or object references on page 83
- Tips for working with change labels on page 83
- Creating and editing change labels on page 84
- Displaying content of change labels on page 89

Assigning change labels when saving in the Designer

To assign changes to a change label in the Designer

- In the Designer, select the **Database > Commit to database** menu item.

- In the toolbar, select the required change label in the **Change label** list. When the changes are transferred to the One Identity Manager database, the change label is applied to the changes affected.

- To assign individual changes to different change labels, use the **Special change label** context menu entry in the change log.
- Save the changes.

Related topics

- Snapshots of objects or object references on page 83
- Tips for working with change labels on page 83
- Assigning objects and dependencies to change labels on page 86
- Editing change labels on page 87

Displaying content of change labels

To display the contents of a change label

1. In the Manager or the Designer, select the **Database > Change management** menu item.

2. In the **Change management** dialog, select the relevant change label in the **Change label** menu.

   If you used this change label to take snapshots of objects, the objects assigned to the change label are shown in the **Tagged changes** pane. The following functions are available:

   - To search within a change label, use **Ctrl +F**.

   - To restrict the information displayed to a single change label, click the arrow in the table header of a column and enter a filter text.

   - Use the context menu to change the order of the changes within a change label. This order is taken into account when the changes are transported.

   - The content of a change for an object is defined in XML format. It specifies whether a property is created, changed, or deleted with a change. To display an XML definition of a change, select **Edit change data**.

   If change labels reference objects, in the **Assigned objects** pane, you can see all the objects that are assigned to the change label.

   TIP: In the Designer, you will find an overview of change labels in the **Base data > General > Change label** category.

Related topics

- Tips for working with change labels on page 83
- Creating and editing change labels on page 84
- Editing change labels on page 87

ONE IDENTITY
by Quest

Locking change labels

If a change label is locked, no further changes can be booked to this label. For example, change labels can be locked and closed after they have been transported with the Database Transporter.

To lock a change label

1. In the Manager or the Designer, select the **Database > Change management** menu item.

2. In the **Change management** dialog next to the **Change labels** menu, click 🔼.

3. Select a change label from the list and open the edit view using ☑.

4. Select the value **True** for the **Locked** property.

5. Click the 💾 button.

6. Click **OK**.

Related topics

- Tips for working with change labels on page 83
- Exporting change labels on page 110

Deleting change labels

You can delete a change label if you do not need it anymore.

To delete a change label

1. In the Manager or the Designer, select the **Database > Change management** menu item.

2. In the **Change management** dialog next to the **Change labels** menu, click 🔼.

3. In the **Change label** dialog, select the change label and click the button 🗑.

4. Confirm the security prompt with **Yes**.

5. To close the **Change label** dialog, click **Cancel**.

6. To close the **Change management** dialog, click **OK**.

Related topics

- Tips for working with change labels on page 83

Release management

You can combine several change labels into one release. There is a report that provides you with an overview of the changes in a release.

To combine change labels into one release

1. In the Designer, select the **Base data > General > Release management** category.

2. Select the **Object > New** menu item.

3. In the edit view, enter a minimum of the following information in the edit view of the **Properties** tab.

   - **Change label**: Enter the name of the change label.
   - **Label type**: Select the **Release** type.

4. In the edit view, select the **Change label** tab and assign the change labels you want to combine into one release.

To display a report about a release

1. In the Designer, select the **Basic data > General > Release management > <name of release>** category.

2. Select the **Show report** task.

   This opens the **Change management release overview** report.

Related topics

- Creating and editing change labels on page 84

Checking data consistency

The consistency check provides different tests for analyzing data objects and to ascertain the current state of their data. In addition to predefined tests, you can define your own tests and, if necessary, run a repair.

You should run a consistency check at regular intervals, as well as after significant changes to the system configuration.

You can run consistency checks in the Manager and in the Designer. The following special cases apply:

- Database tests are run in their entirety in the Manager and the Designer.
- Table tests and object tests in the Manager check the application model data.
- Table tests and object tests in the Designer check the data of the system data model.

Detailed information about this topic

Notes on the consistency check

- It is recommended to run consistency checks with an administrative user.

  Not all consistency checks are available for end users and configuration users.
- To use the Consistency Editor, users need the **Option to call a consistency check for a database** (Common_ConsistencyCheck) program function.
- To use the repair function in the Consistency Editor, users need the **Option to start automatic consistency check repair function** (Common_ConsistencyCheck_Repair) program function.

- Consistency checks of type **Object test** are always run in the context of the user currently logged in. If users do not have any permissions for a certain object, errors may not be identified or repairing errors may fail.

Running consistency checks

To run a consistency check

1. in the Designer or the Manager, start the Consistency Editor with the **Database > Check data consistency** menu item.

   During start up, One Identity Manager schema table definitions are loaded and database objects are made available for testing.

2. Specify the test settings.

   a. In the Consistency Editor toolbar, click .

   b. Enable the test that is to be run and adjust the test settings further if necessary.

   c. Click **OK**.

   NOTE: In the Designer, the test settings dialog opens immediately after the Consistency Editor is started.

3. Start the consistency check. The following test procedures are available in the Consistency Editor for this:

   - Checking all test objects

     Use the ▶ button to start the check.

     NOTE: To exclude individual test objects from the check, use the **Disable** item in the context menu to disable these test objects in the list view before the check starts.

   - Checking single test objects

     In the list view, select the relevant test objects and start this check by selecting **Test**.

     TIP: Use **Shift + select** or **Ctrl + select** to select more than one test object to be checked.

   NOTE: To stop a check that is in progress, click ■ in the Consistency Editor toolbar.

4. Verify error output.

5. Repair errors if necessary.

Related topics

- Notes on the consistency check on page 92
- Displaying test objects and test status on page 95
- Test settings for consistency checks on page 94

Test settings for consistency checks

Define the valid test settings before you run a consistency check. Tests are performed at database, table, and object level. There are already predefined tests available. You can run your own custom tests.

To configure the settings for testing

1. in the Designer or the Manager, start the Consistency Editor with the **Database > Check data consistency** menu item.

2. In the Consistency Editor toolbar, click 👷.

3. Enable the test that is to be run and adjust the test settings further if necessary.

4. Click **OK**.

The tests are grouped according to different criteria.

Table 30: Meanings of the icons used for test settings

Icon	Meaning
🔧	Tests are grouped by themes.
⬇	Tests are grouped by types (database, tables , objects).
⬇	Tests are displayed as a list.
⚡	Tests are grouped by module association.
❌	Tests with **Error** severity are displayed.
⚠	Tests with **Warning** severity are displayed.
ℹ	Tests with **Information** severity are displayed.

Use user-defined tests to run your own tests. You can use the scripts from the script library for these tests. All scripts in the script library are provided for custom tests. The method call of these scripts corresponds to the following syntax.

Database test

```
Public Sub Methodname (ByRef con As IConnection)
Public Sub Methodname (ByVal con As IConnection)
```

Table test

```
Public Sub Methodname (ByRef dbTable As ITableDef)
```

```
Public Sub Methodname (ByVal dbTable As ITableDef)
```

Object test

```
Public Sub Methodname (ByRef dbObject As ISingleDBObject)
```

```
Public Sub Methodname (ByVal dbObject As ISingleDBObject)
```

For detailed information about scripts and the script library, see the *One Identity Manager Configuration Guide*.

Displaying test objects and test status

When Consistency Editor is starting up, One Identity Manager schema table definitions are loaded and database objects are made available for testing. The database tables, the number of objects per table and the test status are displayed in the Consistency Editor's list view.

TIP: To sort by a specific column, click on that column in the table header.

Figure 10: Consistency Editor with Initialized Data
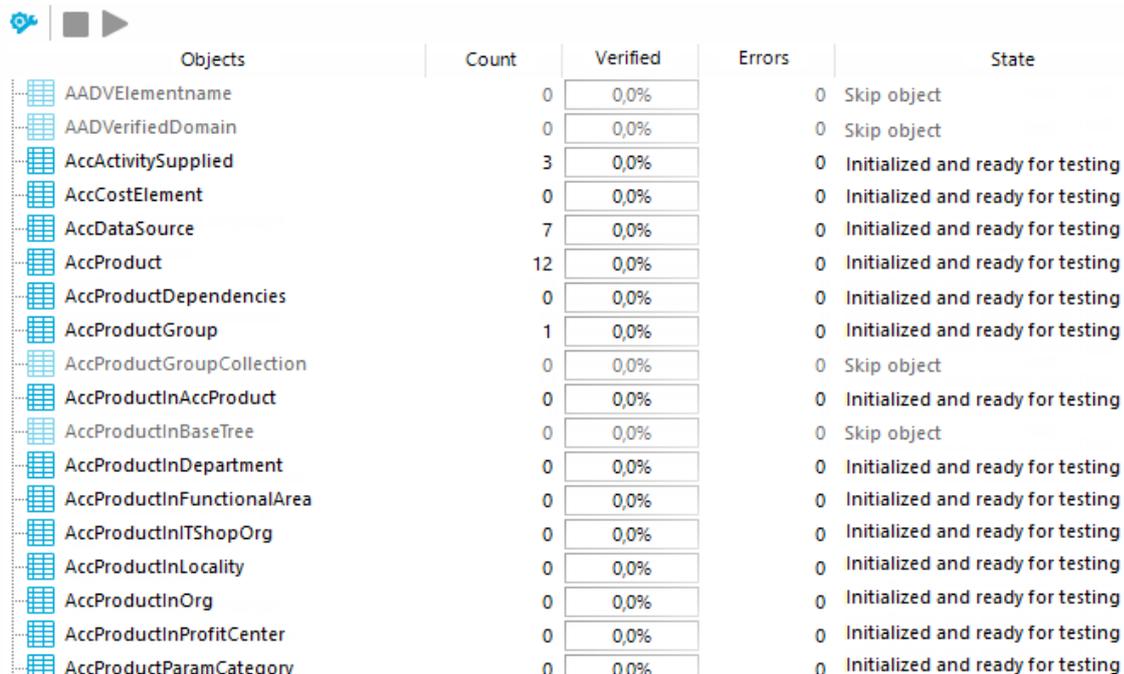
Objects	Count	Verified	Errors	State
AADVElementname	0	0,0%	0	Skip object
AADVerifiedDomain	0	0,0%	0	Skip object
AccActivitySupplied	3	0,0%	0	Initialized and ready for testing
AccCostElement	0	0,0%	0	Initialized and ready for testing
AccDataSource	7	0,0%	0	Initialized and ready for testing
AccProduct	12	0,0%	0	Initialized and ready for testing
AccProductDependencies	0	0,0%	0	Initialized and ready for testing
AccProductGroup	1	0,0%	0	Initialized and ready for testing
AccProductGroupCollection	0	0,0%	0	Skip object
AccProductInAccProduct	0	0,0%	0	Initialized and ready for testing
AccProductInBaseTree	0	0,0%	0	Skip object
AccProductInDepartment	0	0,0%	0	Initialized and ready for testing
AccProductInFunctionalArea	0	0,0%	0	Initialized and ready for testing
AccProductInITShopOrg	0	0,0%	0	Initialized and ready for testing
AccProductInLocality	0	0,0%	0	Initialized and ready for testing
AccProductInOrg	0	0,0%	0	Initialized and ready for testing
AccProductInProfitCenter	0	0,0%	0	Initialized and ready for testing
AccProductParamCategory	0	0,0%	0	Initialized and ready for testing

Table 31: List view information

Column	Meaning
Object	Test object name.
Count	Total number of objects in the database table.
Verified	Test progress in percent.
Errors	The number of error that occurred during a consistency check.
Status	Current test status. The status is updated during the consistency check.

Table 32: Meaning of list view icons

Icon	Meaning
🔍	Test object is currently being test.
✔	Consistency check was successful for this Test object.
✖	Consistency check for this test object is complete but errors occurred.

Table 33: List view context menu items

Context menu item	Meaning
Enable	Enables selected test object(s) for the period of the consistency check.
Disable	Disables selected test object(s) for the period of the consistency check.
Test	Starts running the consistency check for the selected test object (s).
Skip	Skip the test object during the consistency check.

Logging test results

During the consistency check, the number of tested objects and the test status is updated in the editor's list view. Once the test has completed, any error messages are outputted to the Consistency Editor error log.

Table 34: Meaning of icons in the error log

Icon	Meaning
📋	Shows all error messages.

Icon	Meaning
🗒	Only shows errors in the selected objects list view.
❌	A full description of the error is shown in a separate window.
🛠	Fixes the error.
💾	Saves the error messages in a log file.
🗑	Deletes the error messages.

TIP: For a detailed description of an error, double-click the error message.

Related topics

- Repairing errors on page 97

Repairing errors

If automatic error correction is possible, the Consistency Editor error log offers a **Repair** button.

To correct faulty data

1. Select the error entry in the Consistency Editor error log.

   TIP: Use **Shift + select** or **Ctrl + select** to select several entries for repair.

2. To start error correction, click **Repair**.

The correction is made directly in the One Identity Manager database. Resulting data changes are made using the One Identity Manager Service.

NOTE: When repairing templates, dependent objects can also be changed. In certain cases, a large number of dependent objects are changed and saved. Additional processes may be generated.

Related topics

- Notes on the consistency check on page 92

Compiling a One Identity Manager database

After changes have been made to configuration data, such as changes to processes, scripts, templates, object definitions, task definitions or preprocessor-relevant configuration parameters, you must compile the database with the Database Compiler.

After a schema installation, a schema update or the import of a complete custom configuration package, the compilation from the Configuration Wizard or the Database Transporter is started immediately. After importing hotfix packages or restricted custom configuration packages, compile the database using the Database Compiler.

NOTE: The icon in the program's status bar indicates that the database needs to be compiled.

Detailed information about this topic

- Compiling a database with the Database Compiler on page 98
- Output of errors and warnings during compilation on page 100

Compiling a database with the Database Compiler

Before you begin the compilation, all the DBQueue Processor tasks have to be processed. If there are still outstanding tasks on the database, you are notified by the Database Compiler.

To ensure that HTML applications are successfully compiled, you must download packages from the NPM repository. Ensure that the workstation you are compiling on, can establish a connection to the registry.npmjs.org:443 website.

Alternatively, you can download packages from a proxy server and install them manually.

To compile a database

1. In the Designer, select the **Database > Compile database** menu item.

2. On the Database Compiler home page, click **Next**.

3. On the **Compilation settings** page, you can specify which parts of the database are to be recompiled.

Table 35: Compilation settings

Setting	Description
Web services	One Identity Manager offers the option of linking in data that comes from different web service interfaces. The web service proxy code is stored in the database. The Database Compiler compiles the proxy code for all web services of a DLL and saves it in the database. When changes are made to proxy code the database needs to be compiled.
Type-safe database model	Type-safe classes are created from table and column definition that you can use in scripts. As a result, a check whether the correct classes are used is performed when the scripts are written and compiled. TIP: After a schema extension, use this option to compile the database.
Scripts in the Script Library	To compile scripts from the script library, select the following items: • **Do not compile scripts**: The scripts in the script library are not compiled. • **Scripts without dependencies**: This variant results in script changes only becoming effective when the One Identity Manager tools are restarted. • **Scripts including all dependencies**: The scripts and all dependencies, such as templates, tasks, and processes, are recompiled. This guarantees that the script changes are loaded and become effective immediately. One Identity Manager tools do not need to be restarted.
Templates, tasks, etc.	Script expressions, such as templates, formatting scripts, or task definitions are compiled. To limit which code snippets are to be compiled, use 🔽 to show other selection options.
Processes	Processes are compiled. To limit which processes are to be compiled, use 🔽 to show selection options. • **All processes**: Compiles all processes. • **Changed processes**: Compiles all processes that have been modified since the last compilation are compiled. • **Selected processes**: Select single objects whose processes are to be compiled.

Setting	Description
	To select single objects
	1. Click the [**...**] button.
	2. Choose between compiling modified processes, all processes or selected custom processes.
	3. (Optional) Filter the selection further.
	4. Click **OK**.
Web project compilation	Compiles web projects. To limit which web projects are to be compiled, use ▼ to show other selection options.
Compiling the API projects	Compiles API projects.
HTML applications	Compiles HTML applications. To limit which HTML applications are compiled, use ▼ to show other selection options.
Extract language-dependent texts	Texts from scripts are extracted for translation into other languages. The templates are generated for the translation.

4. To start compiling, click **Next**.

5. The compiling progress is displayed on the **Compiling** page. Compiling may take some time. After you close compiling, click **Next**.

6. To end the program, click **Finish** on the last page.

Output of errors and warnings during compilation

If compiler errors or warnings occur:

1. Correct the error after compilation is finished.

2. Recompile the database.

Errors are displayed in a separate log window during the compilation process in the Database Compiler.

- Double-click an error message in the lower part of the log window to jump to the relevant line in the source code view in the upper part of the log window. You can only view the source code you cannot edit it.

- Select **Save** to save the error messages to a file.
- Select **Close** to close the error log. Then the compilation continues.

Figure 11: Error message log
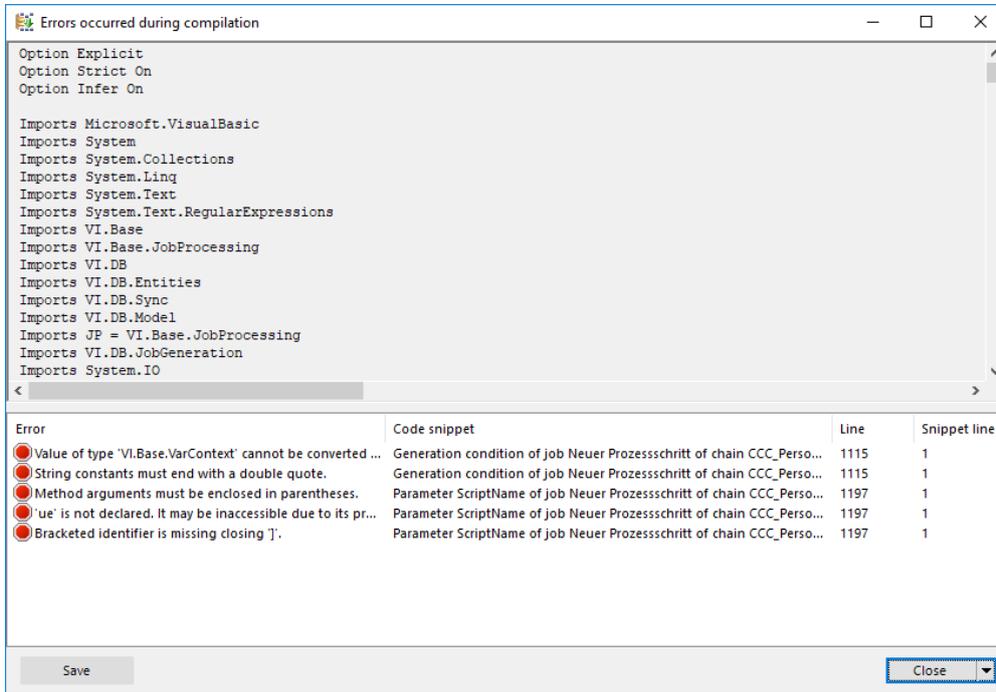
All compiler errors and warnings are recorded during compilation. You can view compiler errors and warnings after compilation is complete.

To display and save messages

- Select the **Show** button to display a message in the error message window. For detailed information about the error message window, see the *One Identity Manager Process Monitoring and Troubleshooting Guide*.

- To save all messages to a file, select an entry and then select **Save log to file** from the context menu.

- To add a message to the clipboard, select the entry and press **Ctrl + C**.

Transporting custom changes

Automatic version control is integrated into One Identity Manager, ensuring that One Identity Manager components are always consistent with each other and with the database. If program extensions that change the structure are implemented - for example, table extensions - the database needs to be updated.

You need to update the database if hotfixes and service packs are available for the version of One Identity Manager you are currently running or for complete version updates. In addition, customer-specific changes must be transferred from a development database into the test database and into the production system database.

Detailed information about this topic

Types of transport packages

You can customize the One Identity Manager schema by loading so-called transport packages. One Identity Manager recognizes the following types of transport packages that can be copied to the database depending on requirements.

Table 36: Transport package

Transport package type	Description	Tool used
Migration package	Migration packages are provided by for the initial database schema installation, for service pack and complete version updates. A migration package contains all the necessary tables, data types, database procedures, and the default One Identity Manager configuration.	Configuration Wizard
Hotfix package	Hotfix packages are provided to load individual corrections to the default configuration such as templates, scripts, processes, or files into the database. NOTE: If a hotfix package only contains changed files, load these files into the database using the Software Loader file.	Database Transporter Software Loader
Custom configuration package	A custom configuration package is used to exchange customer specific changes between the development, test, and productive system database. This transport package is created by the customer and loaded into the database.	Database Transporter

NOTE: If, in additional to a hotfix package, there are additional customized configuration settings to be installed in a One Identity Manager database, create a custom configuration package and use the Database Transporter to import it into the target database. There is no support for merging a hotfix package with a custom configuration package into one transport package.

Related topics

Basics for transporting modifications

Different methods are implemented for transporting modifications.

- Transport of single objects is done through the object layer.

  When you import a transport package, the permissions, templates, and customizer in the target database are taken into account.

This method is used, for example, if you use the Database Transporter program to create and import custom configuration packages that contain modifications to a system user, modifications starting from a defined date or to individual objects.

- The transport of the entire system configuration is done through a transfer buffer.

  All relevant tables are checked when creating the transport package. The condition applied to the table, defines which objects are transported. The primary key is used to establish whether the transport entry has a GUID module and whether it is transferred to the source database transfer buffer. The transfer buffer is read and transport package is created. When importing into the target database, the contents of the transport package is transferred to the target database's transfer buffer. The information is then transferred to the target tables.

  This method is used if you use the Database Transporter program to create and import custom configuration packages that contain the complete system configuration. This method is also used to install and update the One Identity Manager schema using the Configuration Wizard.

When a transport package is imported into a One Identity Manager database, the following operations are carried out:

- Inserting objects

  If no object was found in the destination database using the primary key or alternative key, a new object is created with this key value.

- Updating objects

  An object found in the target database using the primary key will be updated. The update is done using the configuration buffer.

  If transporting modifies a default configuration, the default configuration is moved into the configuration buffer. You can retrieve changes from the configuration buffer and restore the default configuration in this way.

  If, during a One Identity Manager version upgrade, the default configuration is changed by a service pack, a complete version upgrade or by loading a hotfix package, a check is made to see if it has already been customized. In this case, the modified default configuration is copied to the configuration buffer. This ensures that customizations do not go missing.

- Deleting objects

  Objects that are no longer needed are deleted. This operation is always run if the entire system configuration is transported.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107
- Importing transport packages with the Database Transporter on page 116

General tips for transporting changes

To exchange customizations between the development database, test database and the productive database, use the Database Transporter to create transport packages. You also use the Database Transporter to import the transport packages into the target database.

Tips for creating transport packages

- To copy individual objects into a transport package, specify the export criteria in Database Transporter. For example, you can export all changes made by a system user, changes made starting from a defined date or change labels. We recommend that you limit the custom configuration package if you are transporting individual changes.

- You should only create a transport for the full system configuration if you want to copy all the adjustments to the system configuration from a test database into an initial productive database.

- To import transport packages with the Database Transporter, the user needs the program function **Allows transport packages to be imported into the database** (`Transport_Import`).

- The export date, the export description, database revision and the name of the export file in the source database transport history are recorded when a transport package is created with the Database Transporter.

Tips for importing transport packages

- Test the changes in a test environment before you load a transport package in a production system.

- You can display the contents of a transport package with the Database Transporter before you import.

- Before importing a transport package, you can protect individual properties from being overwritten in the target database.

- To import transport packages with Database Transporter, the user requires the **Allows transport packages to be imported into the database** (`Transport_Import`) program function.

- Start Database Transporter on an administrative workstation.

- Depending on the type of transport, the database is set to single-user mode for the duration of the import. Close all existing connections to the database before starting the import.

- When you import a transport package with schema extensions, the database is set to maintenance mode. Objects cannot be processed in the database during this time.

- When importing a transport of the system configuration into a target database, you must also follow the Tips for importing the system configuration on page 115.

- When you import a transport package with the Database Transporter, the import date and description, the database version, and the transport package name are recorded in the transport history of the target database.

Related topics

- Protecting individual properties from being overwritten on page 106
- Displaying transport history on page 106
- Creating transport packages with the Database Transporter on page 107
- Importing transport packages with the Database Transporter on page 116
- Displaying contents of transport packages on page 118

Protecting individual properties from being overwritten

Before importing a transport package, you can protect individual properties from being overwritten in the target database.

For example, you may want to block processing, as follows:

- Configuration parameters and their values should not be overwritten when a test environment is transported to a productive system.
- Server configurations should neither be overwritten in the test environment nor the productive system during a transport.

To unlock and unlock a single property

1. Open the object in the Designer or the Manager.
2. Click the property name and select one of the following options from the context menu:
   - **Prohibit modification**: The property is locked for editing. The input field is locked and grayed-out.
   - **Permit modification:** The property is unlocked and available for editing.

Displaying transport history

The export date, the export description, database revision and the name of the export file in the source database transport history are recorded when a transport package is created with the Database Transporter.

When you import a transport package with the Database Transporter, the import date and description, the database version, and the transport package name are recorded in the transport history of the target database.

To display transport history

- Start the Designer and select the **Help > Transport history** menu item.

Creating transport packages with the Database Transporter

You create transport packages to exchange customer-specific changes between the development, test, and productive system databases.

To create a transport package

1. Start the Launchpad and log in to the One Identity Manager database.
2. In the **Change & Extend** view, select the **Transport custom modifications** entry and click **Start**.

   This starts the Database Transporter program.
3. Select **Create a transport file** on the start page.
4. On the **Select the database connection** page, check the One Identity Manager database connection data and change it if necessary.
5. Enter the information about the transport file on the **Define file name** page.
   a. Enter the name of the transport file and change the output directory as required.
   b. (Optional) To create a log file for the export, set the **Create a log file for data export** option.

      The log file is saved in the output directory of the transport file.
6. Enter a description of the transport data on the **Show and define transport parameters** page.
7. Select the export criteria for the transport on the **Define transport data** page.

   NOTE: You can combine multiple export criteria.

   Table 37: Export criteria

Export criterion	Description
Run SQL statements before data import	You can integrate SQL statements in the custom configuration package, which are to be run before a data import.
	For more information, see Integrating SQL statements in transport packages on page 109.
Transport of favorite objects	In an initial selection, all modified processes, scripts, reports, and mail templates for a specific timeframe are

Export criterion	Description
	offered.
	For more information, see Exporting favorite objects on page 110.
Transport by change label	Transport the changes to objects or object attributes that are summarized in a change label.
	For more information, see Exporting change labels on page 110.
Transport by change information	Limit the transportation data by user, timeframe, and database tables.
	For more information, see Exporting changes based on change information on page 111.
Transport schema extensions	Transport custom schema extensions, such as tables, columns, database procedures, functions, triggers, views, and indexes.
	For more information, see Transporting schema extensions on page 112.
Transporting selected objects and their dependencies	Select single objects and their dependencies for transport.
	For more information, see Exporting selected objects and dependencies on page 113.
Transport system configuration	Transport the entire system configuration.
	For more information, see Transporting the system configuration on page 114 and Tips for importing the system configuration on page 115.
Transport system files	Transport single files.
	For more information, see Exporting system files on page 114.
Transport synchronization projects	Select the synchronization project for transporting.
	For more information, see the *One Identity Manager Target System Synchronization Reference Guide*.
Run SQL statements after the data import	You can integrate SQL statements in the custom configuration package, which are to be run after a data import.
	For more information, see Integrating SQL statements in transport packages on page 109.
Use a transport template for selection	Select a transport template for the transport. The export criteria are determined from the transport template.
	For more information, see Using transport templates on page 119.

8. To start the export, click **Next**.

This create the transport package. The program determines the data to export and displays the progress of the export in the dialog box. The export procedure can take some time.

9. On the last page, the name and storage path for the transport package are displayed. You can open the directory path using the link.

10. To end the program, click **Finish** on the last page.

Related topics

- General tips for transporting changes on page 105
- Importing transport packages with the Database Transporter on page 116
- DBTransporterCMD.exe on page 165

Integrating SQL statements in transport packages

You can integrate SQL statements in the custom configuration package. The SQL statements are run before or after a data import. For example, after a schema extension has been transported a SQL statement may be required for filling initial data in the new columns.

NOTE: To create transport packages with SQL statements, users need the **Enables integration of SQL statements in a transport file** (Transport_SQL) program function.

To run SQL statements within a transport package

1. In the Database Transporter, on the **Define transport data** page, select the export criteria for running SQL expressions. The following export criteria are available:

- Run SQL statements before data import
- Run SQL statements after the data import

2. Create the SQL statement using the **Edit** button. Differentiate between SQL statements for system data transport and user data transport.

a. Enter the SQL statement directly.

- OR -

Use the 📁 button to load a `.sql` file that contains the statements.

b. Use the 💾 button to save to a file.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107

Exporting favorite objects

Use this transport method to select the modified processes, scripts, reports, and mail templates from a specific timeframe.

To transport favorite objects

1. In the Database Transporter, on the **Define transport data** page, select the **Transport of favorite objects** export criteria.
2. Click **Select** to select the single objects for the transport.
   a. In the **Object modified in last ... days** input field, enter the timeframe for the object selection.

   All objects with a change date and user in the selected timeframe are displayed.

   TIP: To include other processes, scripts, reports, or mail templates in the transport package, use the **Load all** entry.

   b. Select the object you want and use ➜ to add it to the transport package.

   TIP: Use **Shift + select** or **Ctrl + select** to select multiple objects in the selection dialog.

   The **Objects to transport** pane lists all selected objects and their dependencies.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107

Exporting change labels

Several changes to objects or objects properties are grouped together under a change label and can be swapped between source database and target database in this way. When a custom configuration package is imported with change labels, new data records are added to the target database and existing data records are updated. In addition, objects marked for deletion in the change label are deleted from the target database.

NOTE: There are no change labels available after initial schema installation.

To transport by change label

1. In the Database Transporter, on the **Define transport data** page, select the **Transport by change information** export criteria.

2. Select the change label from the menu.

3. (Optional) To display the contents of a change label, click **Display**.

   Objects and changes are displayed, which belong to the change label.

   NOTE: If a change label still contains references to objects that no longer exist in the database, remove the assignment using the **Repair** button.

4. (Optional) For additional settings for change label transport, click **Options** and specify the following options.

Table 38: Additional transport settings

Setting	Description
Close change label after export	The change label is completed and locked after transport. No more changes can be booked to this change label.
Copy dependent objects to the transport package	Objects that are dependent on the selected object and do not have a change label are also copied to the transport.
Also display closed change labels	Change labels that are already closed are also offered for selection.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107
- Working with change labels on page 82

Exporting changes based on change information

Use transport by change information to limit transportation data by user, time period and database tables.

To transport by change information

1. In the Database Transporter, on the **Define transport data** page, select the **Transport by change information** export criteria.

2. Specify which changes you want to transport.

Table 39: User list

Entry	Description
me	Only the changes by the logged-in user are added.
all users	Changes are added from all users.
selected users	Changes are added from selected users.
	TIP: The **User** area displays the system users. The **...** button beside the input field allows you to select other users. Use **Shift + select** or **Ctrl + select** to select multiple users in the selection dialog.

3. Use the date filter to export changes for the selected user(s) from a specified date. The entries **today**, **yesterday**, **day before yesterday**, **this week** and **last database migration** and **time period** are available.

4. You can limit transportation data even further by selecting database tables.

5. **Table 40: Table selection**

Entry	Description
Entire system	Changes are added from all tables.
System data	Changes are added from the tables of the system data part.
User data	Changes are added from the tables of the user data part.
Selected tables	Changes are added from specific tables.

TIP: To display objects that match the specified export criteria, click **Display**. In this overview, you can exclude individual objects from the transport To do this, disable the corresponding objects.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107

Transporting schema extensions

Custom schema extensions, like tables, columns, database procedures, functions, triggers, views, and indexes that you want to add, must distinguished by a custom prefix **CCC_**. Furthermore, only custom database procedures, functions, triggers, views, and indexes that are not encoded and are smaller than 64 kb are included.

Custom database procedures, functions, triggers, and views are always exported in their entirety. Entries corresponding to custom tables and columns are generated in the One

Identity Manager schema when the transport package is imported (tables `DialogTable`, `DialogColumn`, `QBMRelation`).

To transport all schema extensions completely from a test database to a productive database, the following procedure is recommended:

1. Create a transport of schema extensions in the test database and import these into the production database.

2. Create a transport of the system configuration in the test database and import these into the production database.

Use the transport options to transport single customizations by change label, change information or selected objects.

To transport schema extensions

- In the Database Transporter, on the **Define transport data** page, select the **Transport of schema extensions** export criteria.

NOTE: Use **Show** to display the schema extensions.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107

Exporting selected objects and dependencies

Use this transport method to select single objects and their dependencies for the transport. You can add objects dependent on the object you want to transport without having to select them individually.

NOTE: The selection for this transport criterion displays all tables not labeled with the **No DB Transport** option. If objects of other tables are to be transportable, then disable the option for the tables in the Designer. For more information about customizing table definitions, see the *One Identity Manager Configuration Guide*.

To transport single objects and their dependencies

1. In the Database Transporter, on the **Define transport data** page, select the **Transport of selected objects and dependencies** export criteria.

2. Click the **Select** button to select the single objects for the transport.

   a. In the **Tables** pane, select the database table from which you want to copy objects to the custom configuration package.

   b. The **Relations** pane displays the ChildRelation (CR), ForeignKey (FK) and many-to-many relations for the selected database table. Enable the required

relations to copy the connected objects to the transport.

    c. The **Objects** pane displays all the objects of the selected table. Select the objects you want and add them to the transport.

- To delete superfluous objects when the transport package is imported, select ⊞.
- If you do not want to perform post-processing when the transport package is imported, select ⊞.

> TIP:
> - Use **Shift + select** or **Ctrl + select** to select multiple objects in the selection dialog.
> - You can use ▼ to create a filter to limit the selection.

    d. The **Objects to transport** pane lists all selected objects and their dependencies.

> TIP: To remove individual object from the transport, select **Remove**.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107

Exporting system files

Use this transport method to transport individual files by exporting them from the database.

To transport new or modified One Identity Manager files

1. In the Database Transporter, on the **Define transport data** page, select the **Transport system files** export criteria.
2. Click **Select** and specify the files to transport.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107

Transporting the system configuration

You should only use a transport of the system configuration if you want to copy all the adjustments to a test database into an initial productive database.

To transport custom database procedures, features, triggers, or views completely from a test database to a productive database in addition to the system configuration:

1. Create a transport of schema extensions in the test database and import these into the production database.
2. Create a transport of the system configuration in the test database and import these into the production database.

To transport individual configuration data units to an existing productive database, use transports based on change labels, change information or selected objects.

Importing a transport of the system configuration overwrites the configuration data of the target database. This also applies to the configuration parameter settings. Before importing a transport package, you can protect individual properties from being overwritten. After importing the system configuration into a target database, you should check and, if necessary, modify the configuration settings.

Detailed information about this topic

- General tips for transporting changes on page 105
- Exporting the system configuration on page 115
- Tips for importing the system configuration on page 115
- Transporting schema extensions on page 112

Exporting the system configuration

You should only use a transport of the system configuration if you want to copy all the adjustments to a test database into an initial productive database.

To create a transport for the system configuration

- In the Database Transporter, on the **Define transport data** page, select the **Transport by change information** export criteria.

Related topics

- General tips for transporting changes on page 105
- Creating transport packages with the Database Transporter on page 107
- Transporting the system configuration on page 114
- Tips for importing the system configuration on page 115

Tips for importing the system configuration

When importing a transport of the system configuration into a target database, you must follow the instructions described under General tips for transporting changes on page 105

and consider the following special features:

- Before performing the import, protect individual properties of the target database from being overwritten.

- If you need custom schema extensions, such as database procedures, features, triggers, or views in the target database in addition to the system configuration, you should import these schema extensions before importing the system configuration.

- After importing the system configuration, check the configuration settings in the target database.

  - Check the staging level of the target database.

  - Check at least the configuration settings for the DBQueue Processor. The settings are specified through the database staging level and configuration parameters.

  You can find detailed information about configuring a One Identity Manager database for test, development, or productive environments in the *One Identity Manager Installation Guide*.

- After importing the system configuration, release the locked properties for editing again.

Related topics

- Exporting the system configuration on page 115
- Protecting individual properties from being overwritten on page 106
- Importing transport packages with the Database Transporter on page 116

Importing transport packages with the Database Transporter

IMPORTANT: Test changes in a test system before you load a transport package in a productive system.

To import a transport package

1. Start the Launchpad and log in to the One Identity Manager database.

2. In the **Change & Extend** view, select the **Transport custom modifications** entry and click **Start**.

   This starts the Database Transporter program.

3. Select **Import transport file** on the home page.

4. On the **Select the database connection** page, check the One Identity Manager database connection data and change it if necessary.

5. Select the transport package file browser and click **Open**.

6. Specify your import options on **Select transport file**.

Table 41: Import options

Setting	Description
Create a log file for the data import	Enable this option to create a log file for the import. The log file is saved in the output directory of the transport file.
Import objects singly and ignore errors	Enable this option to import options individually. Errors, which might occur during importing are ignored and displayed when importing is complete. If you do not enable this option, the import procedure is canceled when errors occur.
Ignore default data differences	Enable this option to ignore changes to default data during the import. If you do not enable this option, the import procedure is canceled if changes to default data are included.

7. Import steps and import progress are displayed on the **Importing transport data** page. The import procedure can take some time. Calculation tasks are queued for the DBQueue Processor on termination.

   NOTE: During import, if the expected value does not match with the actual value in the database, the **Merge conflict** dialog opens. For each conflict, you must decide which value is committed to the database.

   - If you want to keep database value, enable **Current database value**.
   - If you want the value from the transport package to overwrite the database value, enable **Transport value**.

8. If changes have been made to the system configuration, for example, processes, or scripts imported, you have to compile the database after the tasks have been processed. Compilation is started automatically once importing is complete.

9. To end the program, click **Finish** on the last page.

   NOTE: Use the 🖫 button to save any errors that occur whilst importing.

Related topics

- General tips for transporting changes on page 105
- Tips for importing the system configuration on page 115
- Displaying contents of transport packages on page 118

Displaying contents of transport packages

To display the contents of a transport package

1. Start the Launchpad and log in to the One Identity Manager database.

2. In the **Change & Extend** view, select the **Transport custom modifications** entry and click **Start**.

   This starts the Database Transporter program.

3. Select **Show transport file**.

4. Select the transport package file browser and click **Open**.

5. Click **Next** on the **Select transport file** page.

6. The contents of the transport file are displayed on the **Show transport file** page.

   - To display the sequence in which the objects are imported

     1. Click **+** to select an entry in the transport file and select **Sort in import order** from the context menu.

     2. Click **OK** and enter the connection credentials for the database. This step is only required when you established the first in the order.

        The order in which the entry's objects are imported into the database is found.

     3. Repeat this step for all other entries for which you want to determine the import order.

   - To display the objects required for an import in the target environment, select the entry for the `.xml` file and select **Show required objects** from the context menu.

     Objects that are dependent on another object that is not part of the transport package are highlighted.

7. To end the program, click **Finish** on the last page.

TIP: You can start the import of the transport package from display mode. On the **Show transport file** page, click the name of the transport package and use the **Import** context menu.

Related topics

Using transport templates

If you repeatedly need to create transports with certain export criteria, you can use transport templates. You create transport templates in the Database Transporter. Configure a transport and mark it as a transport template. This creates a transport template in XML format. You can use the transport templates when you create transport packages with the Database Transporter or with the `DBTransporterCMD.exe` command line program. This loads the export criteria from the transport template file.

To create a transport template

1. Start the Launchpad and log in to the One Identity Manager database.

2. In the **Change & Extend** view, select the **Transport custom modifications** entry and click **Start**.

   This starts the Database Transporter program.

3. Select **Create a transport file** on the start page.

4. On the **Select the database connection** page, check the One Identity Manager database connection data and change it if necessary.

5. Enter the information about the transport file on the **Define file name** page.

   a. Enter the name of the transport file and change the output directory as required.

   b. Set the **Create a transport template file** option.

   c. (Optional) To create a log file for the export, set the **Create a log file for data export** option.

      The log file is saved in the output directory of the transport file.

6. Enter a description of the transport data on the **Show and define transport parameters** page.

7. Select the export criteria for the transport on the **Define transport data** page.

8. On the **Transport template file** page, enter the following information.

   - **Template file name**: Transport template file name.
   - **Export path for template file**: Repository for the transport template.
   - **Description for template file**: Detailed description of the transport template.

9. To start the export, click **Next**.

   This creates the transport template file and the transport package. The program determines the data to export and displays the progress of the export in the dialog box. The export procedure can take some time.

10. On the last page, the name and storage path for the transport package are displayed. You can open the directory path using the link.

11. To end the program, click **Finish** on the last page.

Related topics

- DBTransporterCMD.exe on page 165
- Creating transport packages with the Database Transporter on page 107

Importing data with the Data Import

With the Data Import program, the One Identity Manager offers a simple means of importing data from other systems. The program supports importing from `.csv` files and importing directly from other database systems. You can import data immediately. You also have the option to import data from customized processes using the import scripts that are created. The import definition is saved so that you can use it for future data imports.

The steps in the program are as follows:

1. Load export definitions

2. Select the import method

3. Configure the import

4. Create an import definition

5. Create an import script

6. Start the import

NOTE:

- For regular data imports into One Identity Manager, you can also use the `ScriptComponent` process component.

- The `DataImporterCMD.exe` program provides support for imports from the command line.

Detailed information about this topic

- Importing data from a CSV file on page 122
- Importing data from an external database on page 127
- Configuring an import on page 130
- Using an import definition file on page 136
- Importing the data on page 135
- DataImporterCMD.exe on page 168

Importing data from a CSV file

Prerequisites

The data structure of the import file needs to fulfill the following requirements:

- The data is separated by a delimiter or fixed column widths are used.
- The data records are separated by a new line.
- Data that contains a new line is marked with a text qualifier.
- For more extensive CSV imports, the data in the import file is sorted in advance to resolve the object dependencies.

  NOTE: For imports with small amounts of data, use the sorting options of the Data Import.

To import data from CSV files into the One Identity Manager database

1. Start the Launchpad and log in to the One Identity Manager database.
2. On the **Configuration** page, select **Configure data import** and click **Start**.

   This starts the Data Import program.
3. On the Data Import start page, click **Next**.
4. On the **Select the database connection** page, check the One Identity Manager database connection data and change it if necessary.
5. (optional) On the **Loading an import definition file** page, load the import definition file, if available.

   NOTE: Leave this field empty if you want to create a new import definition.
6. On the **Select data source** page, select the **Import CSV file** method.
7. On the **Load import file** page, load the import file and enter additional data about the import file.
8. On the **File structure** page, specify how the file is structured.
9. On the **Defining the line structure** page, specify the structure of the lines.
10. (Optional) On the **Line condition** page, specify a condition for importing lines.
11. Configure the import.

    a. On the **Match target tables and columns** page, assign the data for target tables and target columns of the One Identity Manager database and specify the key columns.

    b. On the **Specify hierarchy** page, specify the data hierarchy for the import.

    c. On the **Handling options for data sets** page, specify options for handling the data.

    d. On the **Connection variables** page, define variables that are set on import.

12. On the **Saving the import definition** page, save the import definition file and the import script.

13. On the **Saving the import definition** page, start the import.

14. On the last page of the Data Import, quit the program or start another import.

Detailed information about this topic

Loading the CSV file

On the **Load import file** page in Data Import, enter the following data about the import file.

Table 42: Import file settings

Property	Description
Import file	Path to the .csv file containing the data to be imported. You can use the **...** button beside the input field to navigate to and open the file.
File encoding	Encoding of the .csv file. Encoding of the character set is determined from the character set on your workstation when the import file is loaded. Change the setting if the file was created with another character set.
File culture	Language used to create the file. The language is required in order to read local character formats correctly, for example, dates.
Time zones	If date and time information is imported, select the time zone of the data. The time zone is required for converting the data to UTC.

Structure of the CSV file

On the **File structure** page in the Data Import, specify how the file is structured.

Table 43: File structure

Property	Description
Number of lines in header	Enter the number of head lines in the .csv file. The header is not imported.
Columns identified by	Indicator for column limits. • Select the **Delimiter** option if the data is separated by a semi-colon, comma, space, tab, pipe, or other character. Specify the line structure. • Select the **Fixed width** option if all the data in the columns has the same length. Specify the line structure.

Detailed information about this topic

- Specifying the line structure for data with delimiters on page 124
- Specifying the line structure for data with a fixed width on page 126

Specifying the line structure for data with delimiters

In Data Import on the **Defining the line structure** page, describe how the line structure is configured. If you have selected the **Columns identified by delimiters** option for the file structure, specify the following settings.

NOTE: The **Line break preview** pane displays the line structure according to the selected settings.

Table 44: line structure

Property	Description
Delimiter	Delimiter used to separate the data in the file. You have the following options: **Semicolon**, **Comma**, **Space**, **Tab**, and **Pipe**. If the data is separated by a different character, select **Other:** and enter the delimiter in the input field next to the menu.
Text qualifier	Character enclosing the column text. This text is treated as one value on import, even is the text contains the delimiter given as above. NOTE: The delimiters are masked by doubling them up. Example:

Property	Description

Delimiter:	Comma (,)
Text qualifier:	Quotation mark (")
Value in file:	"Smith,Bill"
Value after import:	Smith,Bill

Delimiter:	Comma (,)
Text qualifier:	Not given or other character:
Value in file:	"Smith,Bill"
1st value after import:	"Smith
2nd value after import:	Bill"

Property	Description
Mask delimiter by doubling	Specifies whether the data is separated by several of the same delimiters. Data that contains a new line must be marked with a text qualifier. Example:

Delimiter:	Comma (,)
Mask delimiter by doubling:	Enabled
Value in file:	Smith,,Bill
Value after import	Smith,Bill

Delimiter:	Comma (,)
Mask delimiter by doubling:	Not set
Value in file:	Smith,,Bill
1st value after import:	Smith
2nd value after import:	
3rd value after import:	Bill

Property	Description
Multiple values in / delimited by	Specifies whether the import contains a multivalued property column (MVP) and the column should not be imported directly. Individual values are entries in another table and should be linked through a many-to-many table. • Using the menu, specify **Multiple values** in the column in question. • In **Delimited by:** enter the values' delimiter. The column values are split up. A new line is generated for each value

Property	Description

although the rest of the columns remain the same.

Example:

The line

> Hans;Meier;Org1|Org2|Org3

is converted by suitable settings to the import source

> Hans;Meier;Org1
>
> Hans;Meier;Org2
>
> Hans;Meier;Org3

Related topics

- Structure of the CSV file on page 123
- Specifying the line structure for data with a fixed width on page 126

Specifying the line structure for data with a fixed width

In Data Import on the **Defining the line structure** page, describe how the line structure is configured. If you have selected the **Columns identified by fixed width** option for the file structure, specify the width of the columns.

- Click on the ruler in the Data Import preview to set a separation point. A separation mark is inserted.
- When you click again on a fixed separation point, the separation mark is deleted.

Related topics

- Structure of the CSV file on page 123
- Specifying the line structure for data with delimiters on page 124

Defining a condition for the import

To exclude individual data records from the import, you can specify a condition for the lines to be imported on the **Line condition** page in the Data Import.

Format the condition in VB.Net syntax. The columns are accessed with dollar notation. For more information about scripts in One Identity Manager, see the *One Identity Manager Configuration Guide*.

You can access the columns using the column index (0...n).

> **Example:**
>
> Do not import the data record if the first column contains the **OLD** value.
>
> Value = 0<>"OLD"

If a header is defined, you can use the column identifier for access.

> **Example:**
>
> Import the data record if the column with the name NewData contains the **True** value.
>
> Value = $NewData:Bool$

Importing data from an external database

To import data from an external database into the One Identity Manager database

1. Start the Launchpad and log in to the One Identity Manager database.

2. On the **Configuration** page, select **Configure data import** and click **Start**.

   This starts the Data Import program.

3. On the Data Import start page, click **Next**.

4. On the **Select the database connection** page, check the One Identity Manager database connection data and change it if necessary.

5. (optional) On the **Loading an import definition file** page, load the import definition file, if available.

   | NOTE: Leave this field empty if you want to create a new import definition.

6. On the **Select data source** page, select the **Import from database** import method.

7. On the **Select external database** page, specify the connection data to the external database.

8. On the **Select source data** page, formulate the query to determine the data records from the external database.

9. Configure the import.

a. On the **Match target tables and columns** page, assign the data for target tables and target columns of the One Identity Manager database and specify the key columns.

b. On the **Specify hierarchy** page, specify the data hierarchy for the import.

c. On the **Handling options for data sets** page, specify options for handling the data.

d. On the **Connection variables** page, define variables that are set on import.

10. On the **Saving the import definition** page, save the import definition file and the import script.

11. On the **Saving the import definition** page, start the import.

12. On the last page of the Data Import, quit the program or start another import.

Detailed information about this topic

Selecting external databases

In the Data Import on the **Select external database**, specify the connection information. Refer to the documentation of the database provider implemented, for the connection parameters.

To set up a connection with an external database

1. In the **Connection type** section, select the provider of the external database.

   - A list of the various database providers available is shown.

Supported database provides
Odbc Data Provider
OleDb Data Provider
OracleClient Data Provider
SQLClient Data Provider
dotConnector for Oracle
MicrosoftSQL Server Compact Data Provider

   - When you use another database provider, select it using the **...** button next to the input field.

2. In the **Connection data** section, enter the connection data to the external database.

   a. Select the **...** button and enter the connection data.

   b. (Optional) To encrypt the connection data, click 🔒.

   c. To check the connection data, click **Test**.

3. If date and time information is imported, select the time zone of the data in the **Other settings** section. The time zone is required for converting the data to UTC.

Determining source data

Formulate the query determine the data records from the external database in the Data Import on the **Select source data** page.

To determine the data from the external database

- To select the table and columns from the external database directly, activate the **Select source table and columns** option and enter the following information.

   Table 45: Settings for selecting the table and columns

Property	Description
Table	Tables whose content is imported.
Columns	Columns whose content is imported.

Property	Description
	Enter the column relations directly in the input field or use the **...** button to open a dialog window to select the columns.
WHERE clause	Condition to further limit the data to be imported.
Order by	The sort order is required if the data records have to be transferred in a defined sequence, for example, as in hierarchical structures. Format the sort order as a valid order by statement for a database query.

- To determine the data records with a SQL query, enable the **SQL statement** option and formulate the database query in SQL syntax.

Configuring an import

Creating an import configuration includes the following steps:

1. Assigning the data to target tables and columns in the One Identity Manager database.
2. Specifying the data hierarchy for the import.
3. Specifying options for handling the data.
4. Defining variables that are set on import.

Detailed information about this topic

Assigning target tables and target columns

On the **Match target tables and columns** page in Data Import, specify the how the data is stored in the One Identity Manager database.

To assign target table and target columns

1. In the **Target table** section, select the target table into which data is imported.

   TIP: Use the ⚒ button in the **Target table** section to assign the target columns and key automatically. You should always check this suggestion.

   Assigns a column if one is found in the target table whose name matches the name in the source column.

2. In the **Target columns and key** section, specify the mapping of data in the target columns of the table.

   NOTE: If a target column is not yet assigned, **Not assigned** is displayed as a column identifier.

   Click the arrow button beside a column identifier to open the assignment wizard and record the following information for every column.

Table 46: Properties for target columns and keys

Property	Description
Use as a key column	Specifies whether the column is used as a key column.
	More than one key columns can be defined. The data records to import into the database are determined based on key columns. Data records should be uniquely identified with these key columns.
Conversion script	Use the conversion script to modify source column values to match the permitted value of the target column. This is required, for example, if a list of permitted values is defined for the target columns.
	Write the conversion script in VB.Net syntax. You access the values with the variable `value`. Use dollar notation to access the source columns. For more information about scripts in One Identity Manager, see the *One Identity Manager Configuration Guide*.
Target column	Select the target columns to be imported into the data. All columns from the target table are displayed with their data type. Following applies:
	• Compulsory data is labeled with a blue triangle in front of the data type.
	• Columns without sufficient permissions are displayed in gray.
	• Columns, deactivated by preprocessor condition, are not shown.
	TIP:
	• Use the ⚒ button to suggest a column if a column whose identifier matches the designation of the source column is found in the target table. You should always check this suggestion.
	• Use the **Show column captions** option to switch between the display name and technical name of the column.

TIP: In the assignment wizard, you can use the **>** button to switch to the next column. The **Data preview** pane contains a preview of the values.

Related topics

- Inserting columns with fixed values on page 132

Inserting columns with fixed values

In Data Import, you can insert additional columns with fixed values in the data import and import into a defined column.

To insert columns with fixed values

1. In the **Target columns and key** section, click the arrow button beside any column name to open the assignment wizard.

2. Click the **+** button.

3. Enter the value in the **Fixed value** field.

    - OR -

    If the value is to be determined from the values in source columns, enter a conversion script.

4. Assign the target column.

5. Close the system assignment wizard.

Related topics

- Assigning target tables and target columns on page 130

Specifying the data hierarchy

If an import contains data that includes dependencies, you must ensure that the reference targets are processed before the reference sources.

For example, child departments (`Department.UID_Department`) are imported after parent departments (`Department.UID_ParentDepartment`).

NOTE:

- Sorting the data into a hierarchical structure can consume a great deal of memory in the Data Import. Therefore, only use this procedure for imports with small amounts of data.

- For more extensive CSV imports, sort the data in advance in the import file to resolve the object dependencies.

- For extensive imports from external databases, use the Order-by clause to sort the data.

To sort the data in the Data Import hierarchically

1. On the **Specify hierarchy** page, enable the **Sort by hierarchy** option.
2. Select the **Key column** in which the data is mapped, for example, Department.UID_Department.
3. Select the **Parent key column**, for example, Department.UID_ParentDepartment.

Related topics

-

Options for handling records

In the Data Import on the **Handling options for data sets** page, specify how new and existing data records are handled when imported. The import must take several cases into account and respond accordingly in each case. During the import, the data records of the source data are compared with the database entries. You can use a condition to further limit the relevant database entries.

Use the following settings to specify how the data records are processed.

Table 47: Options for handling records

Property	Description
Insert new data set	The data record from the source data does not yet exist in the database. If the option is enabled, the data record is inserted in the database.
Adapting existing records	There is an entry in the database that matches the source data record. If the option is enabled, the data record is updated in the database.
	If multiple entries exist in the database, which match the source data record, an entry is written to the error log.
Delete records that no longer exist	The database contains an entry that is not contained in the source data. If the option is enabled, the entry is deleted from the database.
Limiting the target objects	Use a condition to limit the quantity of relevant database entries. The condition is tested when importing begins.
	There is a wizard available though the 📝 button next to the input field, to help you formulate your condition.
	NOTE: If the **Insert new data set** option is enabled, source data records that do not fall within the area of relevant database entries

Property	Description
	due to the limit are handled as new data records and inserted in the database. Under certain circumstances, this can lead to errors such as duplicate data records.

Example for handling data sets
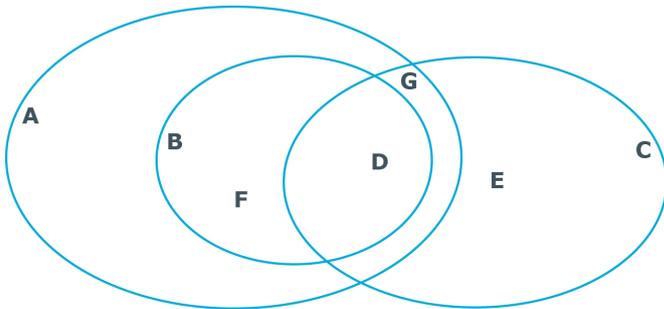
Case	Description
A	All objects in the database.
B	Database set restricted by condition.
C	Entry in source data.
D	All entries in the database and in the source data. Typical action: update all entries in the database.
E	Entries that are only in the source data but not in the database. Typical action: add new entry in the database.
F	Entries that are in the database but not in the source data. Typical action: clean up entries in the database.
G	Entries that are in the source data but no in range selected in the database. These entries are treated as in case E although adding entries may cause conflicts in certain circumstances.

Specifying connection variables

Connection variables are set when the import is run immediately and are also added to the generated import script. You can use the variables in customized processes or templates that are run after importing.

To define a connection variable in the Data Import

1. Click the ✚ button on the **Connection variables** page.

2. Click the **Name** entry and enter the variable name

3. Click the **Value** entry and enter the value of the variables.

To delete a connection variable in the Data Import

- Click the ✖ button on the **Connection variables** page.

Importing the data

The following methods are available to you to import data:

- Start the data import manually in the Data Import. The data records that are processed during import are logged.

- To run data imports on a regular basis, create an import script.

  You can use the import script in custom processes, for example. To create custom processes to run the import, use the DataImport process task of the ScriptComponent process component.

  For more information about creating and editing processes, see the *One Identity Manager Configuration Guide*.

Detailed information about this topic

- Start import immediately on page 135
- Creating import scripts on page 136

Start import immediately

To start the import immediately in the Data Import

1. In Data Import, on the **Saving the import definition** page, set the **Import data** option.

2. To start the import, click **Next**.

   After importing has finished the processing result are displayed. If errors occur during the importing process you can view them with **Show**.

   | TIP: Save the import log using the **Save log as file** context menu item.

Related topics

- Creating import scripts on page 136

ONE IDENTITY
by Quest

Creating import scripts

NOTE: The import script is stored in the One Identity Manager database. To copy import scripts into the database, users need the **Import scripts can be added in the wizard for data import** (DataImport_CreateScript) program function.

To create an import script

1. In Data Import, on the **Saving the import definition** page, enable the **Create import script** option.

2. Enter a name for the import script in **Import script name**.

   Only the VB name are permitted. If a character is not permitted, the field is highlighted in red.

3. Select a change label in **Add script to tag**. Use the **...** button to create a new change label.

4. To create the import script, click **Next**.

5. Compile the script library after saving the script. Click **Yes** to start the compiler.

Related topics

- Start import immediately on page 135
- Working with change labels on page 82

Using an import definition file

The import definition provides you with configuration settings for future data imports. Create the import definition file in the Data Import after creating an import. The import definition is saved as a `.xml` file.

To save an import definition

1. In the Data Import, on the **Saving the import definition** page, enable the **Save import definition file** option.

2. Click the **...** button beside the input field.

3. Select the path and enter the file name.

4. Click **Save**.

Related topics

- DataImporterCMD.exe on page 168

Importing and exporting individual files for the software update

To distribute new or modified files, such as files from a hotfix package or custom form archives, using the automatic software update function to the workstations and servers, import the files into the Software Loader database using the One Identity Manager program.

All files of a One Identity Manager installation are stored in the One Identity Manager database with their name, repository, content, and a hash value. Each file's assignment to the One Identity Manager tools, such as Manager or One Identity Manager Service, is logged.

When you import a file, the Software Loader initially determines the file status based on the file information in the database. To test the file version, the file size and the hash value are determined and compared to the entry in the database.

After a file is successfully imported into the database, the **software revision** semaphore value in the database is updated by the DBQueue Processor. During the next semaphore test, the file is added to the list of files to be updated and is distributed to the workstations and servers.

To equip individual Job servers with the latest software revision manually, you can use the Software Loader program to export individual files from the One Identity Manager database. During the export, the Software Loader checks whether the file already exists in the specified export directory. If this is the case, the file is updated; otherwise, a new version of the file is created.

For detailed information about updating One Identity Manager and about the automatic software update function, see the *One Identity Manager Installation Guide*.

Detailed information about this topic

Importing custom files into One Identity Manager databases

NOTE: When importing custom files, make sure that the directory structure is correctly generated.

- Files for FAT clients do not generally require a subdirectory. When importing the files, select the One Identity Manager installation directory as a base directory.

- Files for web applications generally require a subdirectory, for example a `bin` directory. When importing the files, select the installation directory for the web application as a base directory. This ensures that the necessary subdirectories, such as the `bin` directory, are correctly recognized.

- If a file is required for FAT clients and for web applications, this file must be imported twice; once without a subdirectory and once with a subdirectory.

To import files into a One Identity Manager database

1. Start the Launchpad and log in to the One Identity Manager database.

2. In the **Change & Extend** view, select the **Import files for software update** entry and click **Start**.

    This starts the Software Loader program.

3. Select **Import into database** on the home page.

4. On the **Connect to database** page, check the One Identity Manager database connection data and change if necessary.

5. Specify the file to be imported on **Select files**.

    a. Select the base directory where the files can be found.

    The status and file size of all the files in the selected directory are displayed in the file list.

    Table 48: Meaning of the status

State	Meaning
Version unknown	The file belongs to the known files but has not yet been loaded into the database. There is no version information in the database.
Unknown file	The file is new. The file is in the list of known files but has not been loaded in the database yet. There is no version information in the database.
Version OK	The file version matches the version in the database.
Version modified	The file version has changed with respect to the version in the database.

b. Select the files you want to load into the One Identity Manager database.

> TIP:
> - Click a column in the table header to order the display by the selected column.
> - Press **Shift + select** or **Ctrl + select** to select more than one file.
> - To quickly select all files with **Changed version** as their status, select **Open all directories** and **Open all modified files** in the context menu. Files in subdirectories are only selected if the higher-level directories have already been opened.

6. On the **Select change label** page, assign a change label to make it easier to exchange files between various databases, such as the test database, development database and productive database.

  a. Select **Assign files to following change label**.

  b. Use the button next to the option to select the change label.

7. The files are loaded straight from the One Identity Manager database.

8. Specify other file settings on **Assign machine roles**.

  a. Assign a computer role to the files.

  b. (Optional) For more file settings, click **...** next to the file names.

Table 49: Other file settings

Setting	Description
Source directory	Path to the installation source directory.
Create backup	A copy must be made of the file during the automatic software update.
No update	The file is not updated by the automatic software update.

9. To end the program, click **Finish** on the last page.

Related topics

- Exporting files from a One Identity Manager database on page 140
- Editing file settings for the automatic software update on page 140

Editing file settings for the automatic software update

When importing files using the Software Loader program, you specify whether a backup copy of the existing file is to be created during the automatic software update. You can modify these settings later on.

⚠ **WARNING: Do not change any other file properties as this can lead to errors during the automatic software update.**

To configure the file properties

1. In the Designer, select the **Base Data > Installation > One Identity Manager software** category.

2. Select a file.

3. Edit the following main data.

   - **Create backup**: During the automatic software update, a backup of the existing file is created for files marked with this option.

   - **No update**: Files marked with this option are not updated by the automatic software update.

Related topics

- Importing custom files into One Identity Manager databases on page 138

Exporting files from a One Identity Manager database

To export files from a One Identity Manager database

1. Start the Launchpad and log in to the One Identity Manager database.

2. In the **Change & Extend** view, select the **Import files for software update** entry and click **Start**.

   This starts the Software Loader program.

3. On the home page, select **Export from database**.

4. On the **Connect to database** page, check the One Identity Manager database connection data and change if necessary.

5. Specify which data to export on the **Select files** page.

a. Specify the destination directory to export the file to.

The **Files** tab displays the exported files with their status and file size.

Table 50: Meaning of the status

State	Meaning
Unknown file	The file is not yet exported from the database to the given directory.
Version OK	The file version matches the version in the database.
Version modified	The file version has changed with respect to the version in the database.

b. Mark the files to export.

- To export all the file of a machine role, select the machine role on the **Machine role** tab.

- To select single files, select the files on the **Files** tab.

  TIP:

  - Click a column in the table header to order the display by the selected column.

  - Use **Shift + select** or **Ctrl + select** to select multiple files.

6. The marked files are export to the given directory. This may take some time depending on the number of files selected. The export steps are displayed on the page **Uploading files**. Any export errors are displayed. After exporting is complete, click **Next**.

7. To end the program, click **Finish** on the last page.

Related topics

- Importing custom files into One Identity Manager databases on page 138

Creating a One Identity Manager database for test or development from a database backup

> NOTE: Additional steps are required if you work with granular permissions concepts at SQL level. In this case, contact support. To access the Support Portal, go to https://support.oneidentity.com/identity-manager/.

To create a test or development database from a database backup from another system

1. Create a new database on the database server in the reference environment.

2. Create a database backup of the original database.

3. Restore the backup to the reference database.

4. Restore permissions for the database user on the database server.

5. Compile the database with the Database Compiler.

   Use the Database Compiler to customize connection data for the database and compile all the scripts and processes in the database.

   a. Start the Launchpad and log in to the One Identity Manager database.

   b. In the **Change & Extend** pane, select the **Compile database** item.

      This starts the Database Compiler.

   c. On the Database Compiler start page, click **Next**.

   d. On the **Connect to database** page, enter the connection data for the One Identity Manager database.

   e. This verifies the database ID. If, during the verification, it transpires that the database ID is incorrect, you will be prompted to create a new ID. Confirm this prompt with **OK**. The database ID is changed.

   f. On the **Database connection information insufficient** page, check the database connection data and change if necessary.

- Test the connection parameters (ConnectionString)

  Change the entry using the [**...**] button beside the input field. Select your database connection data.

- Test the full customer name.

g. This test the database connection. Confirm the prompt with **OK**.

h. Enter the valid connection data for the One Identity Manager database.

i. On the **Connect to database** page, enter the connection data for the One Identity Manager database.

j. The components to be compiled are displayed on the **Compilation settings** page. To start compiling, click **Next**.

   This starts the compilation process. The process may take some time.

k. The results of the compilation process are shown on the **Compilation** page. Click **Next** after compilation is complete.

l. Click **Finish** on the last page to end the program.

6. In the Designer, adjust the database staging level of the database.

a. In the Designer, select the **Base Data > General > Databases** category.

b. Select the database and change the value of the **Staging level** property to **Test environment** or **Development system**.

c. Select the **Database > Save to database** and click **Save**.

7. Customize the synchronization project's connection data in the Synchronization Editor.

Initializing DBQueue Processor the after extending the server hardware

After extending the server hardware for the SQL Server, the DBQueue Processor must be initialized again.

To initialize the DBQueue Processor manually

1. Wait until all DBQueue Processor tasks have completed. You can do this using the Job Queue Info.

2. Start a suitable program for running SQL queries and perform the following steps one by one.

   > IMPORTANT: Select a user that you use for migrating the database to run the SQL queries.

3. Check whether other database sessions are active.

   select *

   from sys.sysprocesses p

   where dbid = DB_ID()

   and spid <> @@SPID

   If there are still sessions active, they must be closed first.

4. Stop all DBQueue Processor components.

   exec QBM_PWatchDogPrepare 1

   go

   exec QBM_PDBQueuePrepare 1

   go

5. Initializing the DBQueue Processor.

   exec QBM_PDBQueuePrepare 0,1

   go

   exec QBM_PWatchDogPrepare

   go

For more information about using the DBQueue Processor, see the *One Identity Manager Configuration Guide*.

Command line programs

You can use various command line programs for the automation of One Identity Manager implementations.

Detailed information about this topic

InstallManager.CLI.exe

The `InstallManager.Cli.exe` program provides support for the installation of One Identity Manager. You can run the program from the command line.

IMPORTANT: Run the installation using the command line console in administrator mode.

Calling syntax

`InstallManager.Cli.exe`

```
-m install|change|remove|uninstall

-r {Directory}

[-i {Directory}]

[-fu]

[-mod {ModuleIDs}]

[-d {Targets}]

[-p {Packages}]

[-l {Path}]

[-fo]

[-cs {Service name} {Properties}]

[-dc]
```

Table 51: Program parameters

Parameters	Alternative	Description
-m	--mode	Installation mode. Permitted values are • **install**: Install new modules. • **change**: Update existing modules. • **remove**: Delete modules. • **uninstall**: Uninstall complete installation.
-r	--rootpath	Directory containing the installation sources.
-i	--installpath	(Optional) Directory in which to install.
-fo	--filesonly	(Optional) Only file actions will be run. No start menu entries or registry keys are generated and no services are installed.
-mod	--module	Space-delimited list of module IDs.
-d	--deploymenttarget	Space delimited list of machine roles.
-p	--packages	Space-delimited list of packages.
-l	--logfile	(Optional) Path to the log file.
-fu	--forceupdate	(Optional) All data will be reinstalled.
-cs	--changeservice	Changes the properties for registration of the service. The following values are expected: • **Service name**: Name of the service to be changed • **Properties**: New properties of the service with:

Parameters	Alternative	Description
		- **Name**: Name of the service.
		- **Display**: Display name of the service.
		- **Description**: Description of the service.
		Example:
		`"Name=<New name>;Display=<New display>;Description=<New Description>"`
		You only need to specify the properties that are to be changed.
-dc	--deleteconfig	(Optional) Configuration data and logs are removed in **uninstall** mode.
-h	--help	Display program help.

Example: Installing a single module

```
InstallManager.Cli.exe
          -m install
          -r c:\sourcedir
          -mod QER ADS SAP LDAP ATT
```

Example: Updating a machine role

```
InstallManager.Cli.exe
      -m change
      -r c:\sourcedir
      -d Server\JobServer\ADS
```

Example: Uninstalling the One Identity Manager components

```
InstallManager.Cli.exe
      -m uninstall
      -i c:\installdir
```

```
    -dc
```

DBCompilerCMD.exe

The DBCompilerCMD.exe program supports compiling a database. You can run the program from the command line.

Calling syntax

DBCompilerCMD.exe

    /Conn="{Connection string}"

    /Auth="Module={Authentication string}"

    [/LogLevel=Off|Fatal|Error|Info|Warn|Debug|Trace]

    [-W]

    [/Blacklist=[CompileWebServices] [CompileTypedWrappers] [CompileDialogScripts] [CompileScripts] [CompileJobChains] [CompileWebProjects] [CompileApiProjects] [CompileHtmlApps] [FillMultiLanguage]]

Table 52: Program parameters

Parameter	Description
/Conn	Database connection parameter. Minimum access level **Configuration user**.
	For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*.
	Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
/Auth	Authentication data. The authentication data depends on the authentication module used. For more information about One Identity Manager authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
/LogLevel	(Optional) Scope of output to be processed. Permitted values are: • **Off**: No logging. • **Fatal**: All critical error messages are logged. • **Error**: All error messages are logged.

Parameter	Description
	- **Info**: All information is logged.
	- **Warn**: All warnings are logged.
	- **Debug**: Debugger outputs are logged. This setting should only be used for testing.
	- **Trace**: Highly detailed information is logged. This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome.
-W	(Optional) Wait for the processing of DBQueue Processor tasks to complete before starting compilation.
/Blacklist	(Optional) Space-delimited list of compiler modules that must not be compiled. Permitted values are:
	- **CompileWebServices**: Compile web services
	- **CompileTypedWrappers**: Compile a type-safe database model
	- **CompileDialogScripts**: Compile scripts from the script library
	- **CompileScripts**: Compile templates, formatting scripts and task definitions
	- **CompileJobChains**: Compile processes
	- **CompileWebProjects**: Compile web projects
	- **CompileApiProjects**: Compile API projects
	- **CompileHtmlApps**: Compile HTML applications
	- **FillMultiLanguage**: Extract language-dependent texts
/?	Display program help.

Example:

```
DBCompilerCMD.exe

    /Conn="Data Source=<Database server>;Initial Catalog=<Database name>;User
    ID=<Database user>;Password=<Password>"

    /Auth="Module=DialogUser;User=<User name>;Password=<Password>"

    -W
```

Quantum.MigratorCmd.exe

The Quantum.MigratorCmd.exe program supports migration of a One Identity Manager database. You can run the program from the command line.

Calling syntax for installation

quantum.migratorcmd.exe

/operation=INSTALL

/connection="{Connection string}"

/module={Module IDs}[+]

/destination="{Directory}"

[/password={Password}]

[/admin="mode=create|check|extend;login={SQL Server login name};password={SQL Server login password}]

[/login="User=Config|User;login={SQL Server login name};password={SQL Server login password}"]

[/person="condition={Condition};password={Person password}"]

[/user="DialogUser={System user};password={System user password}"]

[/loglevel="Off|Fatal|Error|Info|Warn|Debug|Trace"]

[/precheck={[+|-] Precheck ID}]

Calling syntax for updating

quantum.migratorcmd.exe

/operation=UPDATE

/connection="{Connection string}"

/module={Module IDs}[+] /destination="{Directory}"

[/admin="mode=create|check|extend;login={SQL Server login name};password={SQL Server login password}]

[/login="User=Config|User;login={SQL Server login name};password={SQL Server login password}"]

[/person="condition={Condition};password={Person password}"]

[/user="DialogUser={System user};password={System user password}"]

[/loglevel="Off|Fatal|Error|Info|Warn|Debug|Trace"]

[/precheck={[+|-] Precheck ID}]

[/keepupdatephase]

Calling syntax for restoring a database

quantum.migratorcmd.exe

> /operation=RESTORE
>
> /connection="{Connection string}"
>
> /destination="{Directory}"
>
> [/admin="mode=create|check|extend;login={SQL Server login name};password={SQL Server login password}]
>
> [/login="User=Config|User;login={SQL Server login name};password={SQL Server login password}"]
>
> [/person="condition={Condition};password={Person password}"]
>
> [/user="DialogUser={System user};password={System user password}"]
>
> [/loglevel="Off|Fatal|Error|Info|Warn|Debug|Trace"]
>
> [/precheck={[+|-] Precheck ID}]
>
> [/keepupdatephase]

Calling syntax for passing parameters as a file

quantum.migratorcmd.exe @File

Table 53: Program parameters

Parameters	Alternative	Description
/operation	-O|-o	Operation to be performed. Permitted values are: • **INSTALL**: This operation installs a new database. • **UPDATE**: This operation updates a database. • **RESTORE**: This operation performs the necessary steps to make the database operational, such as initializing the DBQueue Processor or restoring logins. The operation can be performed after a database has been restored from a backup, for example on another server.
/connection	-C|-c	Database connection parameter. Minimum access level **Administrative user**. For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*. Alternatively, you can enter the name of the

Parameters	Alternative	Description
		connection according to the registry `HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections`.
/module	-M|-m	Comma delimited list of module IDs. For **UPDATE** operation: If the module ID is followed by a plus sign (+), only this module is updated. If no plus sign is specified, all modules listed are updated.
/destination	-D|-d	Source directory .
/password	-P|-p	(Optional) Initial password for the **viadmin** system user when a new database is installed.

/admin		(Optional) Mode for creating SQL Server logins if granular permissions are used.
		
		• **mode**: Specifies in which mode the SQL Server login is created.
		
		  Permitted values are:
		
		    • **create** : The SQL Server login does not exist yet and will be created.
		
		    • **check**: The SQL Server login already exists. This checks the permissions for the SQL Server login. If the necessary permissions are missing, an error message is displayed.
		
		    • **extend**: The SQL Server login already exists. This extends the permissions for the SQL Server login.
		
		• **login**: Name of the SQL Server login.
		
		• **password**: Password for the SQL Server login.
		
		Example: Create a new SQL Server login "OneIM_Admin" with password "secret".
		
		`/admin="mode=create;Login=OneIM_Admin;Password=secret"`

/login		(Optional) Creates the other SQL Server logins if granular permissions are used. Multiple instances of this parameter are possible. • **user**: Name of the database user according to `QBMDBPrincipal.UserName`. Permitted values are:

Parameters	Alternative	Description
		• **Config**: Configuration user.
		• **User**: End user.
		• **login**: Name of the SQL Server login.
		• **password**: Password for the SQL Server login.
		Example: Creating SQL Server logins for configuration users and end users.
		`/login="User=Config;Login=OneIM_Config;Password=secret"`
		`/login="User=User;Login=OneIM_User;Password=secret"`
/person		(Optional) Configuration of an employee's password (`Person.DialogUserPassword`). Multiple instances of this parameter are possible.
		• **condition**: A valid SQL condition for the `person` table.
		• **password**: Password (plain text or password hash).
		Example: Sets the "secret" password for the employee with the internal name "Sys, admin".
		`/person="Condition=InternalName='Sys, admin';Password=secret"`
/user		(Optional) Configuration of a system user's password (`Person.DialogUserPassword`). Multiple instances of this parameter are possible.
		• **DialogUser**: Name of the system user (`DialogUser.UserName`).
		• **password**: Password (plain text or password hash).
/keepupdatephase		(Optional) If the parameter is set, the update phase is not reset to **0** after migration is complete (`DialogDatabase.UpdatePhase`).
/precheck		(Optional) Controls the handling of database pre-checks. The input is given as + or - followed by the ID for the pre-check. Multiple instances of this parameter are possible.
		• +ID: The pre-check is repaired. If the pre-check is not repairable, an error message is displayed.
		• -ID: Pre-check is ignored. This only works for

Parameters	Alternative	Description
		optional tests. An example is `JobqueueEmpty`.
		The ID can be taken from the `PreCheck with ID '{0}' failed!` error message.
/loglevel		(Optional) Scope of output to be processed. Permitted values are: • **Off**: No logging. • **Fatal**: All critical error messages are logged. • **Error**: All error messages are logged. • **Info**: All information is logged. • **Warn**: All warnings are logged. • **Debug**: Debugger outputs are logged. This setting should only be used for testing. • **Trace**: Highly detailed information is logged. This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome.
@file		As an alternative to directly issuing commands, you can name a text file containing the commands. Every command is in a separate line.
/?	-h|-help	Display program help.

Example: Installing a database

quantum.migratorcmd.exe

 /operation=INSTALL

 /connection="Data Source=<Database server>;Initial Catalog=<Database>;User ID=<Database user>;Password=<Password>"

 /module="TSB,ATT,CPL,HDS,POL,RMB,RMS,RPS"

 /destination="C:\install"

Example: Restoring a database

quantum.migratorcmd.exe

```
/operation=RESTORE

/connection="Data Source=<Database server>;Initial Catalog=<Database>;User
ID=<Database user>;Password=<Password>"

/destination="C:\install"

/loglevel=Warning

/precheck=-JobqueueEmpty
```

AppServer.Installer.CMD.exe

The `AppServer.Installer.CMD.exe` program supports installing and uninstalling of application servers. You can run the program from the command line.

| NOTE: Run the installation using the command line console in administrator mode.

Calling syntax for installation

```
AppServer.Installer.CMD.exe

     --conn={Connection string}

     --auth={Authentication string}

     --appname={Application name}

     [--site={site}]

     [--app-pool={Application pool}]

     [--source-dir={Directory}]

     [--deployment-target={Machine role}]

     [--allow-http]

     [--windows-auth]

     [--db-windows-auth]

     [--skip-file-permissions]

     [--runtime-connection={Connection string}]

     [--hdb-connection={History Database ID|Connection string}]

     [/updateuser {User name} [/updateuserdomain {Domain}] [/updateuserpassword
     {Password}]]

     [

          --cert-mode=existing --cert-thumbprint={Thumbprint}

          |
```

```
        --cert-mode=new --cert-issuer {Issuer} [--cert-key=1024|2048|4096]

        |

        --cert-mode=newfile --cert-issuer {Issuer} [--cert-key=1024|2048|4096] [--
        cert-file={Path to certificate file}]

    ]

    [--set-connection]

    [--conn-id={History Database ID}]

    [--verbose]
```

Calling syntax for uninstalling

AppServer.Installer.CMD.exe

    --conn={Connection string}

    --auth={Authentication string}

    --appname={Application name}

    --uninstall

Table 54: Program parameters

Parameters	Alternative	Description
--conn	--connec- tion| -c	Database connection parameter. To install an application server you require at least one user with the **Configuration user** access level.
		For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*.
		Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
--auth	--auth- props|-a	Authentication data for the installation. The authentication data depends on the authentication module used.
		For more information about authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
--appname		Application name.
--site		(Optional) Website on the Internet Information Services where the application is installed. If the parameter is not set, **Default Web Site** is used (default).
--app-pool		(Optional) Application pool. If this parameter is set, the installation is performed in the specified application pool. If this parameter is not set, a new application pool is installed

Parameters	Alternative	Description
		(default).
--source-dir	-s	(Optional) Installation source. If this parameter is set, the installation is performed from the file system. If this parameter is not set, the installation is performed from the database (default).
--deployment-target	-t	(Optional) Machine role for the installation. This parameter can be used more than once. Alternatively, multiple machine role can be separated with a pipe [|]. If this parameter is not set, the **Server | Web | Appserver** machine role is used.
--allow-http		(Optional) If the parameter is set, HTTP is permitted. If this parameter is not available, HTTPS is used (default).
--windows-auth	-w	(Optional) Type of authentication used for the web application. If this parameter is set, Windows authentication is used. If this parameter is not set, anonymous authentication is used on IIS (default).
--db-windows-auth		(Optional) Type of authentication used for the One Identity Manager database. If this parameter is set, Windows authentication is used. If this parameter is not set, the SQL login from the connection parameters is used.
--skip-file-permissions	-f	(Optional) If this parameter is set, no permissions are allocated for the **IIS_USRS** user. If this parameter is not set, the permissions are allocated for the **IIS_USRS** user (default).
--runtime-connection	--run-conn	(Optional) Database connection parameters used as authentication for the One Identity Manager database, for example, if the application server is run with the end user access level. If this parameter is not set, the SQL Server login from the connection parameters is used for the installation (default).
--update-user		(Optional) User for updating. If no user is given, the same user account is used for the application pool.
--update-user-domain		Active Directory domain of the user.
--update-user-password		User password.
--cert-mode		(Optional) Type of certificate selection. Permitted values are:

Parameters	Alternative	Description
		- **existing**: Uses an existing certificate.
		- **new**: Uses a new certificate.
		- **newfile**: Creates a new certificate file. (default)
--cert-thumbprint		Thumbprint of the certificate if an existing certificate is used.
--cert-issuer		Issuer of the certificate if a new certificate or a new certificate file is created. Example: "CN=Application Server"
--cert-key		Length of the certificate's key **1024**, **2048** (default), and **4096** are permitted.
--cert-file		(Optional) Directory path and name of the certificate file if a new certificate file is created. If this parameter is not set, **"App_Data\SessionCertificate.pfx"** is used.
--hdb-connection		(Optional) History Database connection parameter. This value is a combination of the ID and the connection parameter (pipe (|) delimited). Example: "<History Database ID>|key1-1=value1;key2=value2;..."
--set connection	-S	Changes the connection parameters for an installed application.
--conn-id		(Optional) Connection parameter identifier. If this parameter is not set, the application server's own connection parameters are used.
--uninstall	-R	Removes the application server.
--verbose	-v	Detailed log of exception errors.
--help	-h, -?	Display program help.

Parameter formats

Multiple-character options can be given in the following forms:

--conn="..."

--conn "..."

/conn="..."

/conn "..."

Single-character options can be given in the following forms:

-c="..."

-c "..."

```
/c="..."
```

```
/c "..."
```

Switches are allowed in the forms:

```
-R
```

```
/R
```

Example: Installing an application server

```
AppServer.Installer.CMD.exe

    --conn="Data Source=<Database server>;Initial Catalog=<Database name>;User
    ID=<Database user>;Password=<Password>"

    --auth="Module=DialogUser;User=<User name>;Password=<Password>"

    --appname=MyApplicationServer

    --allow-http
```

Example: Uninstalling an application server

```
AppServer.Installer.CMD.exe

    --conn="Data Source=<Database server>;Initial Catalog=<Database name>;User
    ID=<Database user>;Password=<Password>"

    --auth="Module=DialogUser;User=<User name>;Password=<Password>"

    --appname=MyApplicationServer

    --uninstall
```

**Example: Changing the connection parameters of the application
server**

```
AppServer.Installer.CMD.exe

    --set-connection

    --appname=MyApplicationServer

    --conn="Data Source=<Database server>;Initial Catalog=<Database name>;User
    ID=<Database user>;Password=<Password>"
```

> **Example: Changing the parameters for connecting a History Database**
>
> AppServer.Installer.CMD.exe
>
>   --set-connection
>
>   --appname=MyApplicationServer
>
>   --conn-id=<History Database ID>
>
>   --conn="Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>"

AutoUpdate.exe

The `AutoUpate.exe` program supports the automatic software update of a One Identity Manager installation. You can run the program from the command line.

Calling syntax for updating

AutoUpdate.exe

      --conn={Connection string}

      --Install={Directory}

      [--system=MSSQL|APPSERVER]

      [--log-level=Off|Fatal|Error|Info|Warn|Debug|Trace]

Calling syntax for generating a Windows PowerShell start script

AutoUpdate.exe

      --script {path and file name}

Table 55: Program parameters

Parameters	Alternative	Description
--conn	--connection|	Database connection parameter. A user with the minimum permission level **Configuration user** is required.
	-c	For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*.
		Alternatively, you can enter the name of the connection according to the registry `HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections`.

ONE IDENTITY
by Quest

Parameters	Alternative	Description
--install	--install-dir, -i	Installation directory to be updated.
--system	-s	(Optional) Target system for the connection. Permissible values are **MSSQL** (default) or **APPSERVER**.
--script		Creates a Windows PowerShell startup script with the specified file name. The startup script should be located outside the installation directory. The startup script starts the AutoUpdate.exe program and returns when both AutoUpdate.exe and Update.exe are finished.
--log-level	-l	(Optional) Scope of output to be processed. Permitted values are: • **Off**: No logging. • **Fatal**: All critical error messages are logged. • **Error**: All error messages are logged. • **Info**: All information is logged. (default) • **Warn**: All warnings are logged. • **Debug**: Debugger outputs are logged. This setting should only be used for testing. • **Trace**: Highly more information is logged. This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome.
--help	-h, -?	Displays program help.

Parameter formats

Multiple-character options can be given in the following forms:

--conn="..."

--conn "..."

/conn="..."

/conn "..."

Single-character options can be given in the following forms:

-c="..."

-c "..."

/c="..."

/c "..."

Switches are allowed in the forms:

-R

/R

Example: Updating the installation directory

AutoUpdate.exe

    --conn="Data Source=<Database server>;Initial Catalog=<Database name>; User ID=<Database user>; Password=<Password>"

    --install="C:\Temp\InstallFolder"

Example: Updating an installation directory from an application server

AutoUpdate.exe

    --system=APPSERVER

    --conn="URL=https://myappserver/"

    --install="C:\Temp\InstallFolder"

Example: Creating and running a Windows PowerShell startup script

Creating the startup script:

AutoUpdate.exe

    --script=C:\Temp\Update.ps1

Calling the startup script:

.\Update.ps1 -Connection 'Data Source=<Database server>;Initial Catalog=<Database name>; User ID=<Database user>; Password=<Password>' -InstallDir C:\Temp\InstallFolder

---

SoftwareLoaderCMD.exe

Using the SoftwareLoaderCMD.exe program, you can import files into the One Identity Manager database. You can run the program from the command line.

Calling syntax

SoftwareLoaderCMD.exe

```
/Conn="{Connection string}"

/Auth="{Authentication String}"

[/Root="{Path}"]

[-I]

/Files="{files|Targets}"

[-N [-M]]
```

Table 56: Program parameters

Parameter	Description
/Conn	Database connection parameter. Minimum access level **Configuration user**. For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*. Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
/Auth	Authentication data. The authentication data depends on the authentication module used. For more information about authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
/Root	(Optional) File directory.
-I	(Optional) Only in combination with **/Files**. If this parameter is not set, only the files that are already known in the QBMFileRevision table are imported. If this parameter is set, unknown files are also imported into the databased and an entry is created in the QBMFileRevision table.
/Files	List of files with pipe-delimited (|) specification of machine roles that are imported into the database. The entry of wild cards (*) is permitted. Example: /Files="Custom.*.dll|Server|Client"
-N	(Optional) If the parameter exists, all files known in the QBMFileRevision table and located in the directory specified in the **/Root** parameter are updated. **/Conn**, /Auth and /Roo**t are** mand**atory** parameters in this mode. **-I** and /Files are not **taken** into account.
-M	(Optional) If the parameter exists, missing files will be ignored during import. The parameter can be set in connection with parameter **-N.**
-?	Display program help.

> **Example: Updating files that are declared in the QBMFileRevision table.**
>
> SoftwareLoaderCMD.exe
>
> /Conn= "Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>"
>
> /Auth="Module=DialogUser;User=<User name>;Password=<Password>"
>
> /Root="c:\source"
>
> -N

> **Example: Importing custom files**
>
> SoftwareLoaderCMD.exe
>
> /Conn= "Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>"
>
> /Auth="Module=DialogUser;User=<User name>;Password=<Password>"
>
> /Root="c:\customsource"
>
> -I
>
> /Files="Custom.*.dll|Server|Client"

DBTransporterCMD.exe

The DBTransporterCMD.exe program allows you to import transport packages into the One Identity Manager database or export them from the database. You can run the program from the command line.

Calling syntax

DBTransporterCMD.exe

    [-V]

    [-L]

    [-I|-P|-S]

    [-N]

    [-U]

    /File="{Transport file}"

```
/Conn="{Connection string}"

/Auth="{Authentication String}"

[/MergeAction=Error|Transport|Database|Interactive]

[/Options]

[/Template]
```

Table 57: Program parameters

Parameter	Description
/Conn	Database connection parameter. Minimum access level **Configuration user**.
	For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*.
	Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
/Auth	Authentication data. The authentication data depends on the authentication module used. For more information about One Identity Manager authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
/File	Path to the transport file.
/MergeAction	(Optional) Definition of conflict handling. Permitted values are:
	Error: An error occurred. **Transport**: (Default) Values are transferred from the transport package. **Database**: Database values are retained. **Interactive**: User entry for conflict handling expected for each object.
/Options	Special options for importing synchronization projects.
	Example: When transporting a synchronization project, start up configurations, variables and variable sets remain the same in the target database.
	/Options=ObjectFilter=KeepSettings
/Template	Path to the transport template file in XML format. The transport template contains the export criteria. The transport template is used to create the transport file.
	You create transport templates with the Database Transporter. For more information, see Using transport templates on page 119.

Parameter	Description
-V	If this option is set, logging is performed automatically.
-L	If this option is set, a log file is generated for the data import.
-I	If this option is set, errors in insertion and saving are ignored.
-P	If this option is set, errors in insertion are ignored.
-S	If this option is set, saves during insertion are ignored.
-N	If this option is set, the database is not compiled.
-U	If this option is set, the database is not set to single user mode.
-?	Display program help.

Example: Importing a transport package

DBTransporterCMD.exe

    -L

    /File="c:\source\transport.zip"

    /Conn= "Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>"

    /Auth="Module=DialogUser;User=<User name>;Password=<Password>"

Example: Creating a transport package using a template file

DBTransporterCMD.exe

    /Template=c:\data\<Template file>.xml

    /File=c:\data\transport.zip

    /Conn="Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>"

    /Auth="Module=DialogUser;User=<User name>;Password=<Password>"

Related topics

- Creating transport packages with the Database Transporter on page 107

DataImporterCMD.exe

The `DataImporterCMD.exe` program provides support for importing data from CSV files into a One Identity Manager database. You can run the program from the command line. The program requires the import definition files for import. You create import definition files using the Data Import program.

Calling syntax

DataImporterCMD.exe

 /Conn="{Connection string}"

 /Auth="{Authentication String}"

 [/Prov="{Provider}"]

 [/Definition="{Path to import definition file}"]

 [/ImportFile="{path to import file}"]

 [/DefinitionPair="{Path to import definition file}|{path to import file}"]

 [/LogLevel=Off|Fatal|Error|Info|Warn|Debug|Trace]

 [/Culture="{Language code}"]

 [-p]

Table 58: Program parameters

Parameter	Description
/Conn	Database connection parameter. Minimum access level **End user**
	For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*.
	Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
/Auth	Authentication data. The authentication data depends on the authentication module used. For more information about One Identity Manager authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
/Prov	(Optional) Database provider. The following values are permitted: **VI.DB.ViSqlFactory, VI.DB** and **QBM.AppServer.Client.ServiceClientFactory, QBM.AppServer.Client**.
/Definition	Path to the import definition file.
	Example:

Parameter	Description
	`C:\Path\To\Definition.xml`
/ImportFile	Path to the import file. Multiple instances of this parameter are possible. The import definition file specified in the **/Definition** parameter is used.
	Example:
	`C:\Path\To\Import.csv`
/DefinitionPair	Pair of the import definition file and the import file. The files are separated by a pipe character (|). Multiple instances of this parameter are possible.
	Example:
	`C:\Path\To\Definition.xml|C:\Path\To\Import.csv`
/LogLevel	(Optional) Scope of output to be processed. Permitted values are: **Off**: No logging. **Fatal**: All critical error messages are logged. **Error**: All error messages are logged. **Info**: All information is logged. **Warn**: All warnings are logged. **Debug**: Debugger outputs are logged. This setting should only be used for testing. **Trace**: Highly detailed information is logged. This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome.
/Culture	(Optional) Language used to create the file. The language is required in order to read local character formats correctly, for example, dates.
	Example: en-US
-p	(Optional) If this parameter is used, the processing progress is shown.
-?	Display program help.

Example: Importing a single file

`DataImporterCMD.exe`

 `/Prov=VI.DB.ViSqlFactory, VI.DB`

 `/Conn= "Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>"`

```
        /Auth=Module=DialogUserAccountBased

        /Defintion=C:\Work\Import\Data\Def_DataImporter_Employee.xml

        /ImportFile=C:\Work\Import\Data\1_Employees.csv
```

Example: Importing multiple files

```
DataImporterCMD.exe

        /Prov=VI.DB.ViSqlFactory, VI.DB

        /Conn= "Data Source=<Database server>;Initial Catalog=<Database name>;User
        ID=<Database user>;Password=<Password>"

        /Auth=Module=DialogUserAccountBased

        /DefinitionPair=C:\Work\Import\Data\Def_DataImporter_
        Employee.xml|C:\Work\Import\Data\1_Employees.csv

        /DefinitionPair=C:\Work\Import\Data\Def_DataImporter_
        Department.xml|C:\Work\Import\Data\2_Departments.csv

        /DefinitionPair=C:\Work\Import\Data\Def_DataImporter_
        Locality.xml|C:\Work\Import\Data\3_Localities.csv

        /DefinitionPair=C:\Work\Import\Data\Def_DataImporter_
        CostCenter.xml|C:\Work\Import\Data\4_CostCenters.csv
```

Related topics

- Importing data from a CSV file on page 122

SchemaExtensionCmd.exe

The SchemaExtensionCmd.exe program provides support for importing custom schema extensions into a One Identity Manager database.

In databases with a **Test environment** or **Development system** staging level, you can use the program to delete custom schema extensions again.

You can run the program from the command line. The program requires a control file (XML file) for the import. To create control files, use the Schema Extension program. For more information, see the *One Identity Manager Configuration Guide*.

Calling syntax

SchemaExtensionCmd.exe

```
/Conn="{Connection string}"

/Auth="{Authentication String}"

[/Definition="{Path to import definition file}"]

[-f]

[/LogLevel=Off|Fatal|Error|Info|Warn|Debug|Trace]
```

Table 59: Program parameters

Parameter	Description
/Conn	Database connection parameter. Minimum access level **Configuration user**.
	For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*.
	Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
/Auth	Authentication data. The authentication data depends on the authentication module used. For more information about One Identity Manager authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
/Definition	Path to the control file (XML file)
	Example:
	C:\Path\To\Definition.xml
/LogLevel	(Optional) Scope of output to be processed. Permitted values are:
	Off: No logging. **Fatal**: All critical error messages are logged. **Error**: All error messages are logged. **Info**: All information is logged. **Warn**: All warnings are logged. **Debug**: Debugger outputs are logged. This setting should only be used for testing. **Trace**: Highly detailed information is logged. This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome.
-f	(Optional) If this parameter is set, the system does not wait for DBQueue Processor task processing. This can lead to errors if schema extensions are expected that must previously be generated by the DBQueue Processor.
-?	Display program help.

> **Example:**
>
> SchemaExtensionCmd.exe
>
> > /Conn="Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>"
> >
> > /Auth=Module=DialogUserAccountBased
> >
> > /Definition=CustomExtensions.xml

CryptoConfigCMD.exe

The CryptoConfigCMD.exe program supports encryption and decryption of the One Identity Manager database. You can run the program from the command line.

Calling syntax

CryptoConfigCMD.exe

> --conn={Connection string}
>
> --auth={Authentication string}
>
> [--mode=Encrypt|EncryptExisting|Decrypt]
>
> [--private-key= {Path to private key}]
>
> [-y]

Table 60: Program parameters

Parameters	Alternative	Description
--conn	--connection|	Database connection parameter. A user with the minimum permission level **Configuration user** is required.
	-c	For more information about permissions, see the *One Identity Manager Installation Guide* and the *One Identity Manager Authorization and Authentication Guide*.
		Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
--auth	--auth-props|-a	Authentication data for the installation. The authentication data depends on the authentication module used.
		For more information about authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.

Parameters	Alternative	Description
--mode	-m	(Optional) Mode to run. Permitted values are: • **Encrypt**: create a new private key and encrypt the database (default) • **EncryptExisting**: Encrypt the database with an existing key • **Decrypt**: Decrypt the database values.
--private-key	-p	Enter the file with the encryption information. This path must not exist when encrypting a database. The key can be found under this path after the encryption process. This key file must be present when decrypting the database.
-y		(Optional) If the parameter is present, all security queries are answered with **Yes**.
--verbose	-v	Detailed log of exception errors.
--help	-h, -?	Display program help.

Parameter formats

Multiple-character options can be given in the following forms:

--conn="..."

--conn "..."

/conn="..."

/conn "..."

Single-character options can be given in the following forms:

-c="..."

-c "..."

/c="..."

/c "..."

Switches are allowed in the forms:

-R

/R

Example: Encrypt the database with a new key

CryptoConfigCMD.exe

```
      --conn="Data Source=<Database server>;Initial Catalog=<Database name>; User
      ID=<Database user>; Password=<Password>"

      --auth="Module=DialogUser;User=<User name>;Password=<Password>"

      --private-key=C:\path\to\private.key
```

Example: Encrypt the database with an existing key

```
CryptoConfigCMD.exe

      --conn="Data Source=<Database server>;Initial Catalog=<Database name>; User
      ID=<Database user>; Password=<Password>"

      --auth="Module=DialogUser;User=<User name>;Password=<Password>"

      --mode=EncryptExisting
```

Example: Decrypt the database with an existing key

```
CryptoConfigCMD.exe

      --conn="Data Source=<Database server>;Initial Catalog=<Database name>; User
      ID=<Database user>; Password=<Password>"

      --auth="Module=DialogUser;User=<User name>;Password=<Password>"

      --mode=Decrypt

      --private-key=C:\path\to\private.key
```

WebDesigner.InstallerCMD.exe

Using the program `WebDesigner.InstallerCMD.exe`, you can install and uninstall the Web
Portal using the command line console.

NOTE: Run the installation using the command line console in administrator mode.

Calling syntax for installation

```
WebDesigner.InstallerCMD.exe

      [/prov {Provider}]

      /conn {Connection string}
```

```
/authprops {Authentication string}

/appname {Application name}

/site {Site}

[/sourcedir {Directory}]

[/apppool {Application pool}]

[/webproject {Web project}]

[/constauthproj {Subproject name} /constauth {Authentication}]

[/searchserviceurl {url}]

[/applicationtoken {Token}]

[/updateuser {User name} [/updateuserdomain {Domain}]

[/updateuserpassword {Password}]]

[/allowhttp {true|false}]

[-f]

[-w]
```

Calling syntax for uninstalling

```
WebDesigner.InstallerCMD.exe

    [/prov {Provider}]

    /conn {Connection string}

    /authprops {Authentication}

    /appname {Application name}

    [/site {Site}]

    -R
```

Calling syntax for uninstalling earlier Web Portal versions back to and included version 6.x

```
WebDesigner.InstallerCMD.exe

    /appname {Application name}

    [/site {Site}]

    -R
```

Table 61: Program parameters

Parameter	Description
/Prov	(Optional) Database provider – permitted values are **VI.DB.ViSqlFactory, VI.DB** and **QBM.AppServer.Client.ServiceClientFactory,**

Parameter	Description
	QBM.AppServer.Client.
/Conn	Database connection parameter.
	Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
/authprops	Authentication data. The authentication data depends on the authentication module used. For more information about One Identity Manager authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
/appname	Application name.
/site	Internet Information Services web page on which to install the application.
/sourcedir	(Optional) Installation source. If this parameter is set, the installation is performed from the file system. If this parameter is not set, the installation is performed from the database (default).
/apppool	(Optional) Application pool. If this parameter is set, the installation is performed in the specified application pool. If this parameter is not set, a new application pool is installed (default).
/webproject	(Optional) Name of the web project. If this parameter is set, the specified web project is installed. If this parameter is not set, the web project VI_StandardWeb is installed (default).
/constauthproj	Name of the subproject.
/constauth	Authentication settings of the subproject.
/searchserviceurl	Application server for search function availability.
/applicationtoken	Application token for the Password Reset Portal.
/updateuser	(Optional) User for updating. If no user is given, the same user account is used for the application pool.
/updateuserdomain	Active Directory domain of the user.
/updateuserpassword	User password.
/allowhttp	(Optional) If the parameter is set, HTTP is permitted. If this parameter is not available, HTTPS is used (default).
-w	(Optional) Type of authentication used for the web application. If this parameter is set, Windows authentication is used. If this parameter is not set, anonymous authentication is used on IIS (default).

Parameter	Description
-f	(Optional) If this parameter is set, no permissions are allocated for the **IIS_USRS** user. If this parameter is not set, the permissions are allocated for the **IIS_USRS** user (default).
-R	Removes the web application.
/?	Displays program help.

Example: Installation with a direct connection against a SQL Server database.

In this example, the parameters are configured as follows:

- Connection to database on a SQL Server
- Installation in the **default website**
- Application name **testqs**
- Authentication with system user **testadmin**
- Application server for the availability of the search function https://dbserver.testdomain.lan/TestAppServer
- Allow HTTP

WebDesigner.InstallerCMD.exe

```
/conn "Data Source=dbserver.testdomain.lan;Initial
Catalog=IdentityManager;Integrated Security=False;User
ID=admin;Password=password"

/site "Default Web Site"

/appname testqs

/authprops "Module=DialogUser;User=testadmin;Password="

/searchserviceurl https://dbserver.testdomain.lan/TestAppserver

/allowhttp true
```

Example: Installation with a direct connection to an application server

In this example, the parameters are configured as follows:

- Connection to application
- Installation in the **default website**
- Application name **testviaappserver**
- With Windows authentication as web authentication
- User for the updating **User1** with the domain **MyDomain.lan**

```
WebDesigner.InstallerCMD.exe
        /prov "QBM.AppServer.Client.ServiceClientFactory, QBM.AppServer.Client"
        /conn "URL=https://test.lan/IdentityManagerAppServer/"
        /site "Default Web Site"
        /appname testviaappserver
        /authprops "Module=DialogUser;User=testadmin;Password="
        -w
        /updateuser User1
        /updateuserdomain MyDomain.lan
        /updateuserpassword topsecret
```

Example: Uninstalling the web application with a connection against an application server

```
WebDesigner.InstallerCMD.exe
        /prov "QBM.AppServer.Client.ServiceClientFactory, QBM.AppServer.Client"
        /conn "URL=https://test.lan/IdentityManagerAppServer/"
        /appname testviaappserver
        /authprops "Module=DialogUser;User=testadmin;Password="
        -R
```

Example: Processing of authentication settings for a subproject

```
WebDesigner.ConfigFileEditor.exe
        -constAuth ../web.config "test_UserRegistration_Web"
        "Module=DynamicPerson;User[test_USER]=xyz;(Password)Password[test_
```

```
                    Password]=xyz;(Hidden)IgnoreMasterIdentities=;(Hidden)Product=Manager"
```

VI.WebDesigner.CompilerCmd.exe

With the program VI.WebDesigner.CompilerCmd.exe, you can compile the Web Portal using the command line console.

NOTE: Unlike the default settings in the Web Designer, subprojects are not compiled at the same time. This means that when the VI_StandardWeb is compiled, the dI_UserRegistration_Web is not also compiled at the same time.

Calling syntax

VI.WebDesigner.CompilerCmd.exe

> /conn {Connection string}

> /dialog {Authentication string}

> /project {path}

> [/solution {path}]

> [/mode {mode}]

> [-E]

> [-D]

> [-R]

> [/csharpout {folder}]

Table 62: Program parameters

Parameter	Description
/Conn	Database connection parameter.
	Alternatively, you can enter the name of the connection according to the registry HKEY_CURRENT_USER\Software\One Identity\One Identity Manager\Global\Connections.
/dialog	Authentication data. The authentication data depends on the authentication module used. For more information about One Identity Manager authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
/project	Name of the web project.
/solution	(Optional) This parameter specifies the Web Designer solution file to be used. If this parameter is not available, a database project is used.

Parameter	Description
/mode	(Optional) This parameter enables you to specify a compilation mode. Permitted values are: • **normal** : Full compilation (default mode) • **nostore** : No assemblies saved to the database. • **nocompile** : C# code generation runs, but without compilation. • **nocodegen** : Only Web Designer compilation, no C# code generation.
-E	(Optional) This parameter activates the detailed check. For more information about detailed checks, see the *One Identity Manager Web Designer Reference Guide*.
-D	(Optional) This parameter activates the debug compilation.
-R	(Optional) This parameter activates the generation of a stable C# text. This setting prevents use of certain random values.
/csharpout {folder}	(Optional) This parameter contains the target directory for C# text.
/help	Displays program help.

Example: Release compilation of the VI_StandardWeb

VI.WebDesigner.CompilerCmd.exe

/conn "Data Source=<Database server>;Initial Catalog=<Database name>; User ID=<Database user>; Password=<Password>"

/dialog "Module=DialogUser;User=<User name>;Password=<Password>"

/project VI_StandardWeb

Example: Debug compilation of the VI_User_Registration_Web

VI.WebDesigner.CompilerCmd.exe

/conn "Data Source=<Database server>;Initial Catalog=<Database name>; User ID=<Database user>; Password=<Password>"

/dialog "Module=DialogUser;User=<User name>;Password=<Password>"

/project VI_UserRegistration_Web

```
      -D
```

Create-web-dir.exe

With the program `Create-web-dir.exe` you can create a One Identity Manager installation from a directory containing a One Identity Manager setup or from an installed One Identity Manager database. You can find the program on the installation media in the `Modules\QBM\dvd\AddOn\SDK\LinuxWebInstall` directory. You can run the program from the command line.

Calling syntax

create-web-dir.exe

     [--mode=web|standalone]

     --setup={Directory}|[--db-system=MSSQL|APPSERVER]

     --db="{Connection string}"

     --dest={Directory}

     [--modules={Module IDs}]

     --targets= "{Targets}"

     [--nlog={Path}\nlog.config]

     [--web-config={Path}\Web.config]

     [--web-app={URL}]

     [--web-app-project={Web project}]

     [--web-app-product={Product}]

     [--web-app-auth={Authentifier}]

     [--web-app-auth2={Authentifier}]

     [--session-cert={Path}\SessionCertificate.pfx]

     [--create-session-cert]

     [--session-cert-issuer="{Issuer}"]

     [--appserver-url={URL}]

     [--auth="Module={Authentication string}"]

     [--server-name --targets-from-server --config-from-server]

     [--variable{Key=Value}]

     [--nologo]

Calling the program

- Windows: `C:/installer/create-web-dir.exe`

- Linux: `mono /installer/create-web-dir.exe`

Table 63: Program parameters

Parameters	Alternative	Description
--mode		Installation mode. Permitted values are **web** (default) and **standalone**.
--setup	-s	Directory that contains the One Identity Manager setup and from which the installation is to be compiled. Specify either the directory containing the One Identity Manager setup or a database connection.
--db-system		Database system that should be connected as a source. Permissible values are **MSSQL** (default) or **APPSERVER**.
--db	-connect	Connection parameters to the database from which the installation is to be made. Specify either the directory containing the One Identity Manager setup or a database connection.
--dest	-d	Destination directory for the installation.
--targets	-t	Comma-delimited list of machine roles to be installed.
--modules	-m	Comma-delimited list of module IDs for installation. Optional, if the installation is to be done from an existing database.
--nlog	-n	(Optional) Path for the configuration file `NLog.config`.
--web-config	-w	(Optional) Path for configuration file `Web.config`.
--web-app		(Optional) URL for which a web application is to be created in the database.
--web-app-project		(Optional) Name of the web project (`DialogAEDS.FileName`) or UID of the web project (`DialogAEDS.UID_DialogAEDS`).
--web-app-product		(Optional) Name of the product for the web application (`QBMProduct.Ident_Product`) or UID of the product for the web application (`QBMProduct.UID_DialogProduct`).
--web-app-auth		(Optional) Name of the primary authentication module (`DialogAuthentifier.Ident_DialogAuthentifier`) or UID of the primary authentication module (`DialogAuthentifier.UID_ DialogAuthentifier`) for the web application.
--web-app-		(Optional) Name of the secondary authentication module

Parameters	Alternative	Description
auth2		(DialogAuthentifier.Ident_DialogAuthentifier) or UID of the secondary authentication module (DialogAuthentifier.UID_ DialogAuthentifier) for the web application.
--session-cert	-c	(Optional) Path and name of the certificate file.
--create-session-cert		(Optional) Generate a new certificate if the specified certificate file does not exist.
--session-cert-issuer		(Optional) Publisher for the generated certificate file. Standard: **"CN=Application Server"**
--appserver-url		(Optional) URL of the application server to which the web application is to be connected.
--auth		(Optional) Authentication data. The authentication data depends on the authentication module used. For more information about One Identity Manager authentication modules, see the *One Identity Manager Authorization and Authentication Guide*.
--config-from-server		(Optional) Fetches the configuration of the One Identity Manager Service for the Job server from the database and creates the configuration file in the destination directory. If this parameter is set, the **--server-name** parameter is required.
--targets-from-server		(Optional) Fetches the Job server machine roles from the database. If this parameter is set, the **--server-name** parameter is required.
--server-name		(Optional) Name of the Job server in the database (QBMServer table).
--variable	-v	(Optional) Replace the variables in the template files (NLog.config, Web.config) specified with %VariableName%. (Key=Value).
--nologo		(Optional) Specifies whether to display the startup banner and copyright information.
--help	-h, -?	Display program help.

Example: Installing an application server from a directory

```
create-web-dir.exe
    -s=I:\Main\2020.07.28.001
    -d=C:\Work\Install
    -m=ADS,ARS,CAP,CPL,DPR,EBS,EX0,LDP,NDO,QBM,QER,RMB,RMS,SAC,SAP,SBW,SHR,SP0,
    TSB
    -
    t=Server\Web\AppServer,Server\Web\AppServer\SearchCrawler,Server\Web\AppSe
    rver\SearchIndex
    -w=\work\Config\Web.config
    -n=\work\Config\nlog.config
    -c=\work\SessionCertificate.pfx
```

Example: Installing an application server from the database

```
create-web-dir.exe
    -d=C:\Work\Install
    --db-connect="Data Source=<Server>;Initial Catalog=<Database>;User ID=<DB
    User>;Password=<Password>"
    -
    t=Server\Web\AppServer,Server\Web\AppServer\SearchCrawler,Server\Web\AppSe
    rver\SearchIndex
    -w=\work\Config\Web.config
    -n=\work\Config\nlog.config
    -c=\work\SessionCertificate.pfx
```

Example: Installing a Job server from the database with the default machine roles

```
create-web-dir.exe
    --mode=standalone
    -d=C:\Work\Install
```

ONE IDENTITY
by Quest

```
--db-connect="Data Source=<Server>;Initial Catalog=<Database>;User ID=<DB
User>;Password=<Password>"

-t=Server\Jobserver

--server-name=Server1

--targets-from-server
```

Example: Installing a Web Portal from the database

```
mono create-web-dir.exe

      --db-system=MSSQL

      --db-connect='Data Source=<Server>;Initial Catalog=<Database>;User ID=<DB
User>;Password=<Password>'

      --dest=/var/www/OneIMWeb

      --targets="Server\\Web\\EndUserWebPortal"

      --web-app-project="VI_StandardWeb"

      --web-app-auth="RolebasedPerson"

      --appserver-url=http://appserver1.localdomain/

      --web-config=/tmp/web.config
```

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit https://www.oneidentity.com/company/contact-us.aspx.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product

Index

A

application server

    install 156

AppServer.Installer.CMD.exe

    parameter 156

AutoUpdate.exe

    parameter 161

C

calculation schedule

    activation time 52

    configure 47, 49

    enable 49

    runtime interval 49

    set up 47, 49

    start immediately 48

    table 49

    time zone 49

    validity period 49

change label 82-83

    assign 86-88

    assign changes 88

    assign dependencies 86-87

    assign objects 86-87

    close 90, 110

    create 84

    delete 90

    edit 84, 87

    lock 84, 90, 110

    release 91

    remove objects 87

    show contents 89

    transport 110

compile

    database 98

    error message 100

    warning 100

configuration repository 103

consistency check 92

    permission 92

    program function 92

    repair 93

    start 93

    test method 93

    test objects 95

    test setting 94

    test status 95

CryptoConfigCMD.exe

    parameter 172

custom configuration package

    import 116

    show contents 118

D

data export 21

    export definition 23-25

    report 23

    subscribable report 24

data import 121

    assign to employee 132

    change label 135-136

compile 98

E

employee

    locked 81

F

file

    application group 140

    backup 138, 140

    edit 140

    export 140

    import 138

    transport 114

    version 138

H

Hotfix package

    show contents 118

I

info system

    bar chart 36

    configure 34

    diagram type 36

    line diagram 36

    settings 34

    table 36

    tachometer 36

    thermometer 36

    traffic light 36

InstallManager.CLI.exe

    parameter 146

M

mail template 55

    base object 57, 59

    confidentiality 57

    copy 56

    create 56

    design type 57

    edit 56

    email signature 66

    hyperlink 61, 65

    importance 57

    language 57-58

    mail body 57-58

    mail definition 58

    preview 56

    report 57

    subject 57-58

    target format 57

    unsubscribe 57

Mail Template Editor

    preview 56

maintenance task 53

Manager

    apply template 20

    change label 82

    data export 21

    info system 34

    planned operation 15

    process view 41

    simulation mode 9