Copyright 2019 One Identity LLC.
ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of One Identity LLC.

The information in this document is provided in connection with One Identity products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of One Identity LLC products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, ONE IDENTITY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ONE IDENTITY BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ONE IDENTITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. One Identity makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. One Identity does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

One Identity LLC.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (http://www.OneIdentity.com) for regional and international office information.

Patents
One Identity is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at http://www.OneIdentity.com/legal/patents.aspx.

Trademarks
One Identity and the One Identity logo are trademarks and registered trademarks of One Identity LLC in the U.S.A. and other countries. For a complete list of One Identity trademarks, please visit our website at www.OneIdentity.com/legal. All other trademarks are the property of their respective owners.

Legend

⚠️ WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

⚠️ CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

ℹ️ IMPORTANT, NOTE, TIP, MOBILE, or VIDEO: An information icon indicates supporting information.

One Identity Manager Operational Guide
Updated - August 2019
Version - 8.1.1
## Contents

### About this guide

### Simulating data changes in Manager
- Prerequisites for using the simulation mode
- Configuring the simulation report
- Starting and completing a simulation
- Evaluating the simulation data
- Exporting the simulation data

### Planning the times of execution for operations in Manager
- Defining execution times
- Display planned operations
- Restricting the display of scheduled operations
- Labeling input fields with planned changes

### Re-applying templates

### Exporting data with Manager
- Creating a data export
- Saving the export definition as a simple report
- Saving the export definition in a file
- Saving the export definition in the user settings

### Analyzing historical data in TimeTrace
- Prerequisites for displaying the TimeTrace
- Declaring the One Identity Manager History Database for TimeTrace
- Displaying change information in Manager in TimeTrace

### Evaluating process monitoring in the Manager
- Prerequisites for displaying the process information
- Working with the process view
  - Opening the process view
  - Features in the process view
  - Configuring the process view
- Process information layout
## One Identity Manager 8.1.1 Operational Guide

### Layout of logged data changes

### Schedules in One Identity Manager
- Enabling time schedules
- Starting a schedule immediately
- Editing schedules
  - Properties of schedules
- Calculating the time of execution
- Scheduled maintenance tasks

### Mail templates in One Identity Manager
- Creating and editing mail templates
- Copying a mail template
- Creating a mail preview
- General properties of a mail template
- Creating and editing a mail definition
  - Using base object properties
  - Use of hyperlinks for the Web Portal
  - Default functions for creating hyperlinks
  - Using process parameters in hyperlinks
- Customizing email signatures

### Password policies in One Identity Manager
- Predefined password policies
- Using a password policy
- Editing password policies
  - General master data for a password policy
  - Policy settings
  - Character classes for passwords
- Custom scripts for password requirements
  - Script for checking a password
  - Script for generating a password
- Deny list for passwords
- Checking a password
- Testing generation of a password
- Password expiry
- Checking authentication
Displaying locked employees and system users ......................................................... 65

**Working with change labels** .................................................................................. 67
Creating and editing a change label ......................................................................... 67
Displaying contents of a change label ....................................................................... 69
Booking changes to a change label retrospectively ................................................... 69
Deleting a change label .............................................................................................. 71

**Checking data consistency** .................................................................................... 72
Notes on the consistency check .................................................................................. 72
Starting a consistency check ...................................................................................... 73
  Displaying test objects and the test status ............................................................... 74
  Test settings for consistency checks ........................................................................ 75
Logging test results .................................................................................................... 76
Repairing errors ........................................................................................................ 77

**Compiling a One Identity Manager database** ......................................................... 78
Compiling a database with the Database Compiler ................................................... 78
Output of errors and warnings during compilation .................................................... 81

**Transporting custom changes** ................................................................................. 83
Types of transport packages ...................................................................................... 83
Basics for transporting modifications ........................................................................ 84
General notes about transporting changes ................................................................. 86
  Protecting individual properties from being overwritten ....................................... 87
  Displaying transport history .................................................................................... 87
Creating a transport package with the Database Transporter .................................... 88
  Integrating SQL statements in a transport package ............................................... 90
Exporting favorite objects .......................................................................................... 90
Exporting change labels ............................................................................................ 91
Exporting changes based on change information ....................................................... 92
Transporting schema extensions ............................................................................... 93
Exporting selected objects and dependencies ............................................................ 94
Exporting system files ............................................................................................... 95
Transporting the system configuration ...................................................................... 95
  Exporting the system configuration ....................................................................... 96
Notes about importing the system configuration ....................................................... 96
Importing a transport package with the Database Transporter .................................. 97
Importing data with Data Import ................................................. 100
Importing data from a CSV file .................................................. 101
Loading the CSV file .................................................................. 102
Structure of the CSV file ............................................................... 102
  Specifying the line structure for data with delimiters ...................... 103
  Specifying the line structure for data with a fixed width .................... 105
Defining a condition for the import ............................................... 105
Importing data from an external database ........................................ 106
Selecting an external database ....................................................... 107
Determining the source data .......................................................... 108
Configuring an import .................................................................. 108
Assigning the data to target tables and target columns ....................... 109
  Inserting columns with fixed values ............................................ 110
Specifying the data hierarchy ......................................................... 111
Options for handling records ......................................................... 111
Specifying connection variables .................................................... 113
Importing the data ....................................................................... 113
  Start import immediately .......................................................... 114
  Create an import script ............................................................ 114
Using an import definition file ....................................................... 115
Importing and exporting individual files for the software update ........... 116
Importing custom files into a One Identity Manager database ............... 117
  Editing file settings for the automatic software update .................... 119
Exporting files from a One Identity Manager database ........................ 119
Appendix: Command line programs .................................................. 121
InstallManager.CLI.exe ................................................................ 121
DBCompilerCMD.exe ................................................................... 122
Quantum.MigratorCmd.exe ............................................................ 123
WebDesigner.InstallerCMD.exe ....................................................... 124
VI.WebDesigner.CompilerCmd.exe ................................................... 126
SoftwareLoaderCMD.exe ............................................................... 128
DBTransporterCMD.exe .................................................................. 130
DataImporterCMD.exe ................................................................... 131
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 the One Identity Manager.

You will learn how to analyze and monitor data changes in Manager. It describes how you schedule execution 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 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.

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 the One Identity Manager documentation in Manager and in Designer by selecting Help | Search. The online version of the One Identity Manager documentation is available in the Support-Portal under Online-Documentation. You will find videos with additional information at www.YouTube.com/OneIdentity.
Simulating data changes in Manager

Using the simulation mode in 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 on page 9
- Starting and completing a simulation on page 10
- Evaluating the simulation data on page 11
- Exporting the simulation data on page 13
- Configuring the simulation report on page 10

Prerequisites for using the simulation mode

- To use the simulation mode in Manager, the user needs 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 plug-ins in the Manager program settings.
- To ensure that the users can export the simulation data, enable the Common
Simulation | ExportReport configuration parameter in Designer. If necessary, configure the report for exporting the simulation data.

Related topics

- Configuring the simulation report on page 10

Configuring the simulation report

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

**To change the current report:**

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

**Table 1: Available simulation reports**

<table>
<thead>
<tr>
<th>Technical name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VID_DatabaseSimulationResult_Export</td>
<td>The report shows the simulation data without evaluating the rules. This report is the default report.</td>
</tr>
<tr>
<td>VID_DatabaseSimulationResult_with_Compliance_Export</td>
<td>The report shows the simulation data including an evaluation of the rules.</td>
</tr>
</tbody>
</table>

Starting and completing a 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.

ℹ️ **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 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. To do this in
   the Manager, select Object | Save or Object | Specify execution time.

Related topics

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

Evaluating the simulation data

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

Table 2: Logging simulation data

<table>
<thead>
<tr>
<th>View</th>
<th>Description</th>
<th>Displayed information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>This gives you an overview of which actions the applied changes will trigger. You can export the simulation data and display the report.</td>
<td>Number of applied changes for each action.</td>
</tr>
<tr>
<td>DBQueue</td>
<td>The DBQueue log shows the following information. You can show the additional information from the context menu.</td>
<td>Operation Calculation tasks to be run.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sort order Sort order to process the calculation task.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Process ID Unique process ID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Object Unique object ID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Child object Unique ID of the child object.</td>
</tr>
<tr>
<td>View</td>
<td>Description</td>
<td>Displayed information</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Generated process</td>
<td>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.</td>
<td>Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Process steps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Property</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
</tr>
<tr>
<td>Trigger changes</td>
<td>Shows all changes made to objects that have been triggered during the simulation.</td>
<td>Table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Column</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Old value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New value</td>
</tr>
<tr>
<td>Changed object</td>
<td>Shows objects and their properties if they were affected by the changes made during simulation.</td>
<td>Table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Column</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Old value</td>
</tr>
<tr>
<td>View</td>
<td>Description</td>
<td>Displayed information</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Rule evaluation</td>
<td>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.</td>
<td>New value</td>
</tr>
<tr>
<td>Employee</td>
<td>Employee who has newly violated the rule or is no longer violating the rule for the first time.</td>
<td>Employee</td>
</tr>
<tr>
<td>Rule violation</td>
<td>Type of change (rule no longer violated or new rule violation) and the affected rule.</td>
<td>Rule violation</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the rule violation.</td>
<td>Description</td>
</tr>
</tbody>
</table>

**Related topics**

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

**Exporting the simulation data**

*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 report, click **Save**.
5. Click **Yes** to show the report now.

**Related topics**

- Prerequisites for using the simulation mode on page 9
- Configuring the simulation report on page 10
Planning the times of execution for operations in Manager

To run operations at a later time rather than immediately, specify a time of execution in Manager for the individual operations. You can specify times of execution for the default operations such as creating, changing and deleting an object and schedule the execution of custom-defined tasks and events. The DBQueue Processor checks whether planned operations exist. When the planned time is reached, the operation is run.

To enable scheduling of a time of execution

- Check in Designer whether the **Common | DeferredOperation** configuration parameter is enabled. Otherwise, set the configuration parameter and compile the database.
- 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

- Defining execution times on page 15
- Display planned operations on page 15
Defining execution times

To plan a time of execution for creating and changing an object
1. In the Manager, select the object for which you wish to specify an execution time.
2. Select Change master data.
3. Change the values you wish to edit.
4. Select Object | Specify execution time.
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 menu item Object | Set deletion time.
3. Specify the date and time of deletion.
4. Enter additional information on the operation under Remarks.
5. Click Save.

Related topics
- Display planned operations on page 15
- Labeling input fields with planned changes on page 17

Display planned operations

To display all planned operations
- In the Manager menu, click Database | Show deferred operations.

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

**Table 3: Information on data changes**

<table>
<thead>
<tr>
<th>Information</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Name of the table to which the data record belongs. This is used to group the objects.</td>
</tr>
<tr>
<td>Object</td>
<td>Object affected by the operation.</td>
</tr>
<tr>
<td>Operation</td>
<td>Operation that is run for the object. Permitted operations are <strong>Add object</strong>, <strong>Change object</strong>, <strong>Delete object</strong>, <strong>Generate event</strong> and <strong>Call method</strong>.</td>
</tr>
<tr>
<td>Time of execution</td>
<td>Time at which the operation should be run.</td>
</tr>
<tr>
<td>Comment</td>
<td>Additional comment on the operation.</td>
</tr>
<tr>
<td>Created by</td>
<td>User who created the planned operation.</td>
</tr>
</tbody>
</table>

**TIP:** Click a remark to show the remark in full.

**Table 4: Meaning of icons in the form toolbar**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>Load and display the selected object.</td>
</tr>
<tr>
<td>⚪</td>
<td>Execute scheduled operations now</td>
</tr>
<tr>
<td>🗑</td>
<td>Delete selected objects.</td>
</tr>
<tr>
<td>🔄</td>
<td>Re-enable selected objects. If an error occurred during the operation, you can run the change again.</td>
</tr>
<tr>
<td>🔄</td>
<td>Reload the data.</td>
</tr>
<tr>
<td>🔍</td>
<td>Filter view.</td>
</tr>
</tbody>
</table>

**Related topics**

- Restricting the display of scheduled operations on page 17
Restricting the display of scheduled operations

To restrict the information shown 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 menu, click **Database | Show deferred operations**.
2. In the overview of schedule operations, open the **Filter view** menu.
3. Select one or more filters under **State** or **Operation**.

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

<table>
<thead>
<tr>
<th>Filter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td></td>
</tr>
<tr>
<td>Outstanding operations</td>
<td>Shows or hides pending operations.</td>
</tr>
<tr>
<td>Expired operations</td>
<td>Shows or hides operations whose time of execution has already expired.</td>
</tr>
<tr>
<td>Operation</td>
<td></td>
</tr>
<tr>
<td>Create object</td>
<td>Shows or hides all entries with the <strong>Add object</strong> operation.</td>
</tr>
<tr>
<td>Change object</td>
<td>Shows or hides all entries with the <strong>Change object</strong> operation.</td>
</tr>
<tr>
<td>Delete object</td>
<td>Shows or hides all entries with the <strong>Delete object</strong> operation.</td>
</tr>
<tr>
<td>Generate event</td>
<td>Shows or hides all entries with the <strong>Generate event</strong> operation.</td>
</tr>
<tr>
<td>Calling methods</td>
<td>Shows or hides all entries with the <strong>Call method</strong> operation.</td>
</tr>
<tr>
<td>Show all</td>
<td>All scheduled operations are displayed.</td>
</tr>
</tbody>
</table>

Labeling input fields with planned changes

Input fields for which changes are planned at a specific time are labeled in the Manager with additional icons. The new values are not shown for security reasons.
Table 6: Labeling the input fields with scheduled changes

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚨</td>
<td>The value will be changed at a specified time. You can change the value only at the specified time.</td>
</tr>
<tr>
<td>🕒</td>
<td>The value will be changed at a specified time.</td>
</tr>
</tbody>
</table>
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 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 Manager.
- This could also cause large numbers of dependent objects to be modified and processes to be generated.

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 Change master data.
3. In the menu, click Object | Reapply templates.
4. Save the changes.
Exporting data with Manager

Using the Manager, you can export the data for the application data model. An export form supports the export of data in CSV format, which you can edit with Microsoft Office Excel or import into other One Identity Manager databases. 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.

**NOTE:** To export data in Manager, the user needs the Data export option (Common_DATABASEExport) program function.

**Detailed information about this topic**

- Creating a data export on page 20
- Saving the export definition as a simple report on page 21
- Saving the export definition in the user settings on page 23
- Saving the export definition in a file on page 22

**Creating a data export**

*To create an export*

1. In the Manager, use the **Database | Export data** menu item to open the export form.
2. In the **Column selection** area 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 **Export**.

   **TIP:** To mark all columns, use the button in the toolbar. To clear all selected columns, click . You can use the button to display the display names or the technical names.
4. Use the option **Export display value** 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** area, use the ▼, ▲ and ▼ buttons to adjust the sort order of the export columns.

6. (Optional) In the **Condition** area, 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 text box to open the wizard.

7. In the **Export data** area, use the button to create an export preview.

   The data sets that match the export criteria are shown in a table. 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:](image) 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** area, 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**.

   ![NOTE:](image) 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.

**Related topics**

- Saving the export definition in the user settings on page 23
- Saving the export definition in a file on page 22
- Saving the export definition as a simple report on page 21

**Saving the export definition as a simple report**

A simple report with the export definitions is created, which can be displayed and subscribed to in 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 plug-in in the program settings in Manager.

To create a simple report with the export definition

1. In Manager, select Database | Export data 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, see the One Identity Manager Report Subscriptions Administration Guide and the One Identity Manager Web Portal User Guide.

Related topics
- Saving the export definition in the user settings on page 23
- Saving the export definition in a file on page 22
- Creating a data export on page 20

Saving the export definition 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 Manager, select Database | Export data 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 Manager, select **Database | Export data** 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 the export definition in the user settings on page 23
- Saving the export definition as a simple report on page 21
- Creating a data export on page 20

**Saving the export definition in the 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 Manager, select **Database | Export data** 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
To load an export definition from the user settings:
1. In Manager, select Database | Export data 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 the Export name menu.
5. Click Open.

To delete an export definition from the user settings:
1. In Manager, select Database | Export data 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 the export definition in a file on page 22
- Saving the export definition as a simple report on page 21
- Creating a data export on page 20
Analyzing historical data in 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.

Figure 2: Analyzing historical data

Detailed information about this topic

- Prerequisites for displaying the TimeTrace on page 26
- Displaying change information in Manager in TimeTrace on page 27
Prerequisites for displaying the TimeTrace

Prerequisites for using the TimeTrace

- The `Common | ProcessState` configuration parameter is enabled.
- The method for recording changes to data within the process monitoring is configured.
  For detailed information about configuring the recording of data changes, see One Identity Manager Configuration Guide.
- To display archived data in the TimeTrace, the One Identity Manager History Database in the One Identity Manager database must be identified.

  **NOTE:** Changes to data that are stored in the One Identity Manager database can be evaluated immediately.

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

Related topics

- Declaring the One Identity Manager History Database for TimeTrace on page 26

Declaring the One Identity Manager History Database for TimeTrace

In order to include archived data in TimeTrace function of the One Identity Manager, the One Identity Manager History Database databases must be identified in the One Identity Manager database.

**To link a History Database into a TimeTrace**

1. In Designer, select Base Data | General | TimeTrace databases.
2. Select Object | New.
3. Enter the One Identity Manager History Database's name.
4. 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 One Identity Manager History Database.

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Database server.</td>
</tr>
<tr>
<td>Windows authentication</td>
<td>Specifies whether 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.</td>
</tr>
<tr>
<td>User</td>
<td>SQL Server Login name.</td>
</tr>
<tr>
<td>Password</td>
<td>SQL Server login password.</td>
</tr>
<tr>
<td>Database</td>
<td>Database.</td>
</tr>
</tbody>
</table>

5. Select **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.

## Displaying change information in Manager in TimeTrace

**To display an object’s change data:**

1. Open the time trace using the **View** | **TimeTrace** menu in Manager.
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 an 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 master 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 master 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.

   **Table 8: Properties for the transfer of historical data**

<table>
<thead>
<tr>
<th>Information</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>These properties are modified if the historical value is transferred. The changes are made immediately or through templates.</td>
</tr>
<tr>
<td>New value</td>
<td>Value of the property after the historical value was saved.</td>
</tr>
<tr>
<td>Old value</td>
<td>The current property value is displayed. This value is overwritten if you save the historical value.</td>
</tr>
</tbody>
</table>

2. Click **Save**.
Evaluating process monitoring in the Manager

In the 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 the One Identity Manager. For detailed information about the process monitoring methods, see the One Identity Manager Configuration Guide.

In the Manager 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 29
- Working with the process view on page 30
- Opening the process view on page 30
- Process information layout on page 32
- Layout of logged data changes on page 34

Prerequisites for displaying the process information

- The process view in 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 recording 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 Manager, the user needs 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 area 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 area of the process view. You will find 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 29
- Opening the process view on page 30
- Features in the process view on page 31
- Configuring the process display on page 32
- Process information layout on page 32
- Layout of logged data changes on page 34

**Opening the process view**

*To open the process view:*

- In the Manager menu, click View | Process data

**Related topics**

- Prerequisites for displaying the process information on page 29
- Features in the process view on page 31
- Process information layout on page 32
- Layout of logged data changes on page 34
Features in the process view

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>Reload process data.</td>
</tr>
<tr>
<td>🔄</td>
<td>Show process data for the current user (object-related process data).</td>
</tr>
<tr>
<td>🔄</td>
<td>Show process data for the selected object (object-related process data).</td>
</tr>
<tr>
<td>🔄</td>
<td>Show processes for related objects.</td>
</tr>
<tr>
<td>🔄</td>
<td>Show substitute processes.</td>
</tr>
<tr>
<td>🔍</td>
<td>Filter process data by status.</td>
</tr>
<tr>
<td>🔍</td>
<td>Show data changes for the current user in the log (user-specific changes).</td>
</tr>
<tr>
<td>🔍</td>
<td>Show data changes to the object selected in the result list in the log (object-related changes).</td>
</tr>
</tbody>
</table>

Table 10: Items in the process data form context menu

<table>
<thead>
<tr>
<th>Context Menu Item</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>The system searches for objects in the process view.</td>
</tr>
<tr>
<td>Add to favorites</td>
<td>Adds the selected object to your favorites.</td>
</tr>
<tr>
<td>Remove from favorites</td>
<td>Removes the selected object from your favorites.</td>
</tr>
<tr>
<td>Task</td>
<td>The object’s available forms are shown and you can switch to the desired form.</td>
</tr>
<tr>
<td>Object type:&lt;BaseObject&gt;</td>
<td>This shows the base object of the triggered process.</td>
</tr>
<tr>
<td>Show process logs</td>
<td>The log shows the data changes of the selected process (process-related changes).</td>
</tr>
<tr>
<td>Properties</td>
<td>Other properties of the active object are displayed. This menu item is only available in expert mode.</td>
</tr>
</tbody>
</table>

Table 11: Meaning of toolbar icons in the log

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>The selected object appears in the document view.</td>
</tr>
<tr>
<td>🔄</td>
<td>The display switches to the originally referenced (old) object and this is shown in document view.</td>
</tr>
</tbody>
</table>
Icon  Meaning

The display switches to the newly referenced object and this is shown in document view.

Configuring the process display

To configure the process display in Manager

1. In Manager, select the Database | Settings menu item
2. On the Functionality tab, configure the following settings in the Process information area.
   - **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.
     - **Full information**: Single steps with a detail depth of basic information, extended information, and full 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

<table>
<thead>
<tr>
<th>Information</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities</td>
<td>Process data display text for the process.</td>
</tr>
<tr>
<td>Status</td>
<td>Process status.</td>
</tr>
<tr>
<td>Triggered by</td>
<td>User who triggered the process.</td>
</tr>
<tr>
<td>Triggered on</td>
<td>Time of action.</td>
</tr>
<tr>
<td>Duration</td>
<td>Processing time.</td>
</tr>
<tr>
<td>More information</td>
<td>More information on the status, such as attempts to repeat individual</td>
</tr>
<tr>
<td></td>
<td>steps or a start time for deferred steps.</td>
</tr>
<tr>
<td>Process ID</td>
<td>Unique ID (GenProcID). Changes that can be traced back to a single</td>
</tr>
<tr>
<td></td>
<td>cause are given the same Process ID and are grouped in this way.</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>Processing was completed with success (status Finished).</td>
</tr>
<tr>
<td>►</td>
<td>The process is currently being processed (status Active).</td>
</tr>
<tr>
<td>⚠️</td>
<td>An error occurred during processing (status Error).</td>
</tr>
<tr>
<td>✨</td>
<td>Status of processing (status Pending, Delayed, Frozen or Not reached).</td>
</tr>
<tr>
<td>✸</td>
<td>Process dependent on selected process.</td>
</tr>
<tr>
<td>🔴</td>
<td>Previous substitute process.</td>
</tr>
<tr>
<td>🔵</td>
<td>Next substitute process.</td>
</tr>
</tbody>
</table>

Related topics

- Layout of logged data changes on page 34
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 [icon] in the process data form.
- To show all of an object's data changes, select the object in the result list and click [icon] in the process data form.

The data changes log shows the following information.

**Table 14: Information on data changes**

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

**Table 15: Meaning of icons in the log**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌐</td>
<td>Column</td>
</tr>
<tr>
<td>🌐</td>
<td>Table</td>
</tr>
<tr>
<td>🌐</td>
<td>Foreign key</td>
</tr>
<tr>
<td>🌐</td>
<td>Object</td>
</tr>
</tbody>
</table>

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 [icon]. 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 32
• Analyzing historical data in TimeTrace on page 25
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. Schedules are required, for example, for scheduled execution of processes within process handling or for different calculation tasks within One Identity Manager. A schedule can be in control of several tasks. Execution times are configured in a schedule for the tasks to be executed.

You create and edit schedules in Designer or in Manager. The Designer displays all schedules of the system. You can edit individual schedules such as schedules for attestations or schedules for compliance calculations in the Manager. For detailed information about editing schedules in 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 executed. Before you use a schedule on a repeated basis, check the effects of the process handling.

Related topics

- Enabling time schedules on page 36
- Starting a schedule immediately on page 37
- Editing schedules on page 37
- Calculating the time of execution on page 39
- Scheduled maintenance tasks on page 42

Enabling time schedules

For detailed information about editing schedules in Manager, refer to the administration guides for the modules.
**To enable a schedule in the Designer**

1. Select **Base data | General | Schedules** in Designer.
2. Select the schedule.
3. Set **Enabled**.
4. Select **Database | Save to database** and click **Save**.

**Starting a schedule immediately**

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

**NOTE:**
- Before you start a schedule manually, check whether other processes will be executed as a result, that also need to be preprocessed by the One Identity Manager.
- The last execution 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 ▶.
4. Confirm the security prompt with **Yes**.

**Editing schedules**

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

**To edit a schedule in the Designer**

1. In the Designer, select the **Base data | General | Schedules** category.
2. Select the schedule.
   - OR -
   Select **Object | New** to create a new schedule.
3. Edit the schedule’s master data.
4. Select **Database | Save to database** and click **Save**.
Detailed information about this topic

- Properties of schedules on page 38

Properties of schedules

Enter the following properties for a schedule.

**Table 16: Schedule properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Schedule ID. Translate the given text using the button.</td>
</tr>
<tr>
<td>Description</td>
<td>Detailed description of the schedule. Translate the given text using the button.</td>
</tr>
<tr>
<td>Table</td>
<td>Table whose data can be used by the schedule.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Specifies whether the schedule is enabled or not.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> Only active schedules are run.</td>
</tr>
<tr>
<td>Time zones</td>
<td>Unique identifier for the time zone that is used for running the schedule. Choose between <strong>Universal Time Code</strong> or one of the time zones in the menu.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> When you add a new schedule, the time zone is preset to that of the client from which you started the Designer.</td>
</tr>
<tr>
<td>Start (date)</td>
<td>The day on which the schedule should be run for the first time.</td>
</tr>
<tr>
<td>Validity period</td>
<td>Period within which the schedule is run.</td>
</tr>
<tr>
<td></td>
<td>- If the schedule will be run for an unlimited period, select the option <strong>Unlimited duration</strong>.</td>
</tr>
<tr>
<td></td>
<td>- To set a validity period, select the option <strong>Limited duration</strong> and enter the day the schedule will be run for the last time in End (date).</td>
</tr>
<tr>
<td>Occurs</td>
<td>Interval in which the task is run. Permitted interval types are <strong>Every minute</strong>, <strong>Hourly</strong>, <strong>Daily</strong>, <strong>Weekly</strong>, <strong>Monthly</strong> and <strong>Yearly</strong>.</td>
</tr>
<tr>
<td></td>
<td>For the <strong>Weekly</strong> interval type, specify the precise weekday. For the <strong>Monthly</strong> interval type, specify the day of the month (1st to 31st day of the month). For the <strong>Yearly</strong> interval type, specify the day of the year (1st to 366th day of the year).</td>
</tr>
</tbody>
</table>
### Property | Meaning
--- | ---
| **NOTE:** If the schedule is not going to be run until next month because the interval type is Monthly with sub interval 29, 30 or 31, the last day of the current month is used. Example: A schedule that is run on the 31st day of each month is run on 30th April. In February, the schedule is run on the 28th (or 29th in leap year). Schedules with the interval type Yearly with sub interval 366 are only run in leap year. |  

| Start time | Fixed start type for the Daily, Weekly, Monthly and Yearly interval types. Enter the time in local format for the chosen time zone. For the interval types Every minute and Hourly, the start time is calculated from the rate of occurrence and the interval type. |
| Repeat every | Rate of occurrence for running the schedule within the selected time interval. For the Weekly interval type, select at least one weekday. |
| Last planned run/Next planned run | Execution time calculated by the DBQueue Processor. Execution 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:** The 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. |

## Calculating the time of execution

The database schedule QBM_PWatchDog on <database> verifies the schedules that need to be run and their start 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 interval types Every minute and Hourly, the next planned time of execution will be determined from the time at which the database schedule runs, the specified time zone and the execution rate. For schedules with the interval types Daily, Weekly, Monthly and Yearly, the next planned time of execution will be determined from the current day, the specified subinterval and the start time within the specified time zone.
Behavior of new schedules

- The execution times for a new schedule are initially empty.
- When the database schedule is run for the first time after a new schedule is set up, only the last planned time of execution (date 01/02/1900) and the next planned time of execution (date 12/30/1899) are entered and the schedule prepared. The task is not executed.
- The next time the database schedule is run, the next possible time of execution is determined, without taking the execution rate into account. The task is not executed.
- The task is run if the execution time has been reached. When the next scheduled run is calculated, this time the interval will be taken into account.

Behavior of Modified Schedules

- When a schedule is modified, the next scheduled run field is cleared.
- When the database schedule is run for the first time after the schedule is changed, the next possible time of execution is determined, without taking the execution rate into account. The task is not executed.
- The task is run if the execution time has been reached. When the next scheduled run is calculated, this time the interval will be taken into account.

Example 1

Run a schedule every 15 minutes.

<table>
<thead>
<tr>
<th>Current time</th>
<th>Monday, 7/14/2014, 8:59 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval type</td>
<td>Every minute</td>
</tr>
<tr>
<td>Interval</td>
<td>15</td>
</tr>
</tbody>
</table>

Since the times of execution for new schedules are initially empty, the schedule is prepared during the next database schedule run at 09:00 a.m. The following run of the database schedule at 9:01 am determines the next scheduled run.

This results in the follow scenario:

<table>
<thead>
<tr>
<th>Last scheduled run</th>
<th>1/2/1900, 12:00:00 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next scheduled run</td>
<td>7/14/2014, 9:16:00 AM</td>
</tr>
</tbody>
</table>

This task is run for the first time the next time the database schedule runs at 9:16 am. The next scheduled run is calculated as follows:

<table>
<thead>
<tr>
<th>Last scheduled run</th>
<th>7/14/2014, 9:16:00 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next scheduled run</td>
<td>7/14/2014, 9:31:00 AM</td>
</tr>
</tbody>
</table>
Example 2

Run a schedule once a week on Wednesdays at 12 pm.

<table>
<thead>
<tr>
<th>Current time</th>
<th>Monday, 7/14/2014, 8:59 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval type</td>
<td>Weekly</td>
</tr>
<tr>
<td>Interval</td>
<td>1</td>
</tr>
<tr>
<td>Sub interval</td>
<td>Wednesday</td>
</tr>
<tr>
<td>Start time</td>
<td>12:00 PM</td>
</tr>
</tbody>
</table>

Since the times of execution for new schedules are initially empty, the schedule is prepared during the next database schedule run at 09:00 a.m.. The following run of the database schedule at 9:01 am determines the next scheduled run. The next possible execution time is determined based on the current date (07/14/2014) without taking the interval into account.

This results in the following scenario after the database schedule has run:

<table>
<thead>
<tr>
<th>Last scheduled run</th>
<th>1/2/1900, 12:00:00 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next scheduled run</td>
<td>7/16/2014, 12:00:00 PM</td>
</tr>
</tbody>
</table>

The task is executed for the first time when the database schedule is run on 7/16/2014 at 12:00 PM. The next scheduled run is calculated as follows:

<table>
<thead>
<tr>
<th>Last scheduled run</th>
<th>7/16/2014, 12:00:00 PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next scheduled run</td>
<td>7/23/2014, 12:00:00 PM</td>
</tr>
</tbody>
</table>

Example 3

The task should be executed quarterly on the 15th day of the month at 06:00 p.m.

<table>
<thead>
<tr>
<th>Current time</th>
<th>Monday, 7/14/2014, 8:59 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval type</td>
<td>Monthly</td>
</tr>
<tr>
<td>Interval</td>
<td>3</td>
</tr>
<tr>
<td>Sub interval</td>
<td>15</td>
</tr>
<tr>
<td>Start time</td>
<td>6:00 PM</td>
</tr>
</tbody>
</table>

Since the times of execution for new schedules are initially empty, the schedule is prepared during the next database schedule run at 09:00 a.m.. The following run of the database schedule at 9:01 am determines the next scheduled run. The next possible
execution time is determined based on the current date (07/14/2014) without taking the interval into account.

This results in the follow scenario after the database schedule has run:

<table>
<thead>
<tr>
<th>Last scheduled run</th>
<th>1/2/1900. 00:00:00 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next scheduled run</td>
<td>8/15/2014, 6:00:00 PM</td>
</tr>
</tbody>
</table>

The task is executed for the first time when the database schedule is run on 8/15/2014 at 6:00 PM. The next scheduled run is calculated as follows:

<table>
<thead>
<tr>
<th>Last scheduled run</th>
<th>8/15/2014, 6:00:00 PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next scheduled run</td>
<td>11/15/2014, 6:00:00 PM</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Task</th>
<th>Schedule</th>
<th>Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce size of change entries</td>
<td>Reduce logs</td>
<td>Daily</td>
</tr>
<tr>
<td>Reduce size of process tracking logs</td>
<td>Reduce logs</td>
<td>Daily</td>
</tr>
<tr>
<td>Purge dynamic users</td>
<td>Reduce logs</td>
<td>Daily</td>
</tr>
<tr>
<td>Reduce size of process log entries</td>
<td>Reduce logs</td>
<td>Daily</td>
</tr>
<tr>
<td>Reduce size of process history</td>
<td>Reduce logs</td>
<td>Daily</td>
</tr>
<tr>
<td>Populate calendar</td>
<td>Daily maintenance tasks</td>
<td>Daily</td>
</tr>
<tr>
<td>Lock table statistics</td>
<td>Daily maintenance tasks</td>
<td>Daily</td>
</tr>
<tr>
<td>Calculate table statistics</td>
<td>Daily maintenance tasks</td>
<td>Daily</td>
</tr>
<tr>
<td>Rebuild table index</td>
<td>Daily maintenance tasks</td>
<td>Daily</td>
</tr>
<tr>
<td>Delete closed cases in the IT Shop</td>
<td>Daily maintenance tasks</td>
<td>Daily</td>
</tr>
<tr>
<td>Calculate statistics for data contents</td>
<td>Weekly maintenance tasks</td>
<td>Weekly</td>
</tr>
<tr>
<td>Set RowLock</td>
<td>Weekly maintenance tasks</td>
<td>Weekly</td>
</tr>
</tbody>
</table>
Related topics

- Schedules in One Identity Manager on page 36
Mail templates in One Identity Manager

The 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 master data such as target format, important or mail notification confidentiality and one or more mail definitions. Mail text is defined in several languages in the mail template. This ensures that the language of the recipient is taken into account when the email 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 attestations in Manager. For detailed information about editing mail templates in Manager, refer to the administration guides for the modules.

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

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 on page 45
- General properties of a mail template on page 46
- Creating and editing a mail definition on page 47
- Customizing email signatures on page 53
Creating and editing mail templates

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

To edit a mail template in Designer

1. In 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 Designer

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

Related topics

- Copying a mail template on page 45
- Creating a mail preview on page 46

Copying a mail template

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

To copy a mail template in the Designer

1. In 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 Mail template | Copy mail template.
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 45
- Creating a mail preview on page 46
Creating a mail preview

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

To display a mail template preview in the Designer
1. In Designer, select the Mail templates category.
2. Select the mail template and start Mail Template Editor using the Edit mail template task.
3. Select Mail templates | Mail preview.
4. Select the base object and click OK.

General properties of a mail template

Table 18: Mail template properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mail template name</td>
<td>Name of the mail template. This name will be used to display the mail templates in the administration tools and in Web Portal. Translate the given text using the button.</td>
</tr>
<tr>
<td>Base object</td>
<td>Mail template base object. A base object only needs to be entered if the mail definition properties of the base object are referenced.</td>
</tr>
<tr>
<td>Report (parameter set)</td>
<td>Report, made available through the mail template.</td>
</tr>
<tr>
<td>Description</td>
<td>Mail template description. Translate the given text using the button.</td>
</tr>
<tr>
<td>Target format</td>
<td>Format in which to generate email notification. Permitted values are:</td>
</tr>
<tr>
<td></td>
<td>- <strong>HTML</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>- <strong>TXT</strong>: 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.</td>
</tr>
<tr>
<td>Design type</td>
<td>Design in which to generate the email notification. Permitted values are:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Mail template</strong>: The generated email notification contains the mail body in accordance with the mail definition.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Report</strong>: The generated email notification contains the report</td>
</tr>
<tr>
<td>Property</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>specified under <strong>Report (parameter set)</strong> as its mail body.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Mail template, report in attachment</strong>: The generated email notification contains the mail body in accordance with the mail definition.</td>
</tr>
<tr>
<td></td>
<td>The report specified under <strong>Report (parameter set)</strong> is attached to the notification as a PDF file.</td>
</tr>
<tr>
<td>Importance</td>
<td>Importance for the email notification. Permitted values are <strong>Low</strong>, <strong>Normal</strong>, and <strong>High</strong>.</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>Confidentiality for the email notification. Permitted values are <strong>Normal</strong>, <strong>Personal</strong>, <strong>Private</strong>, and <strong>Confidential</strong>.</td>
</tr>
<tr>
<td>Can unsubscribe</td>
<td>Specifies whether the recipient can unsubscribe email notification. If this option is set, the emails can be unsubscribed through the Web Portal.</td>
</tr>
<tr>
<td>Deactivated</td>
<td>Specifies whether this mail template is disabled.</td>
</tr>
<tr>
<td>Mail definition</td>
<td>Unique name for the mail definition.</td>
</tr>
<tr>
<td>Language</td>
<td>Language which applies to the mail template. The recipient’s language preferences are taken into account when an email notification is generated.</td>
</tr>
<tr>
<td>Subject</td>
<td>Subject of the email message</td>
</tr>
<tr>
<td>Mail body</td>
<td>Content of the email message.</td>
</tr>
</tbody>
</table>

**Related topics**

- Creating and editing a mail definition on page 47

### Creating and editing a mail definition

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. Open the mail template in Mail Template Editor.
2. Click the button next to the Mail definition list.
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 Designer, enable the corresponding countries. For more detailed 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. Open the mail template in Mail Template Editor.
2. Select the language in **Mail definition**.
3. Edit the mail subject line and the body text.
4. Save the changes.

**Related topics**

- Creating and editing mail templates on page 45
- Using base object properties on page 48
- Use of hyperlinks for the Web Portal on page 49
- Default functions for creating hyperlinks on page 50
- Using process parameters in hyperlinks on page 52
- Customizing email signatures on page 53

**Using base object properties**

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 detailed 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**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>base object</td>
<td>PersonWantsOrg</td>
</tr>
<tr>
<td>Subject</td>
<td>&quot;$DisplayOrg[D]&quot; status change</td>
</tr>
</tbody>
</table>
| 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 $DateHead:Date$.
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: $DisplayPersonHead[D]$
Reason: $ReasonHead[D]$

Related topics
- Creating and editing a mail definition on page 47

Use of hyperlinks for the Web Portal

You can add hyperlinks to Web Portal in the mail text of a mail definition. If the recipient clicks on the hyperlink in the email, the Web Portal is opened 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 Web Portal. You edit the configuration parameter in Designer.

http://<server name>/<application>
with:
<server name> = name of server
<application> = path to the Web Portal installation directory
To add a hyperlink to 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, which you can use to create hyperlinks in Web Portal.]

3. To accept the input, click OK.

Related topics

- Creating and editing a mail definition on page 47
- Default functions for creating hyperlinks on page 50
- Using process parameters in hyperlinks on page 52

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 script VI_BuildAttestationLinks 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

<table>
<thead>
<tr>
<th>Function</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VI_BuildAttestationLink_Show</td>
<td>Opens the attestation page in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildAttestationLink_Approve</td>
<td>Approves an attestation and opens the attestation page in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildAttestationLink_Deny</td>
<td>Denies an attestation and opens the attestation page in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildAttestationLink_AnswerQuestion</td>
<td>Opens the page for answering a question in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildAttestationLink_Pending</td>
<td>Opens the page with pending attestations in the Web Portal.</td>
</tr>
</tbody>
</table>

Default Functions for IT Shop Requests

The script VI_BuildITShopLinks 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

<table>
<thead>
<tr>
<th>Function</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VI_BuildITShopLink_Show_for_Approver</td>
<td>Opens the overview page for request approval in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildITShopLink_Show_for_Requester</td>
<td>Opens the overview page for requests in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildITShopLink_Approve</td>
<td>Approves a request and opens the approvals page in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildITShopLink_Deny</td>
<td>Denies a request and opens the approvals page in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildITShopLink_AnswerQuestion</td>
<td>Opens the page for answering a question in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildITShopLink_Reject</td>
<td>Opens the page with denied requests in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildAttestationLink_Pending</td>
<td>Opens the page with pending requests in the Web Portal.</td>
</tr>
<tr>
<td>VI_BuildITShopLink_Unsubscribe</td>
<td>Creates the link for canceling email notification. This function is used in processes for unsubscribing email notifications.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Function</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VI_BuildComplianceLink_Show</td>
<td>Opens the exception approval page in the Web Portal.</td>
</tr>
</tbody>
</table>

Default function for policy checking

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

Table 23: Functions of the VI_BuildComplianceLinks script

<table>
<thead>
<tr>
<th>Function</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VI_BuildQERPolicyLink_Show</td>
<td>Opens the exception approval page in the Web Portal.</td>
</tr>
</tbody>
</table>

Related topics

- Creating and editing a mail definition on page 47
- Use of hyperlinks for the Web Portal on page 49
- Using process parameters in hyperlinks on page 52

Using process 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 Process Editor.

Example for populating the 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 in the Address input field:

$PC(<ParamName>)$

Example:

$PC(NoSubscription)$

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

Related topics

- Creating and editing a mail definition on page 47
- Use of hyperlinks for the Web Portal on page 49
- Default functions for creating hyperlinks on page 50

Customizing email signatures

Configure the email signature for mail templates using the following configuration parameter. Edit the configuration parameters in the Designer. The VI_GetRichMailSignature script compiles the email signature according to the configuration parameters for use in mail templates.

Table 24: Configuration Parameters for Email Signatures

<table>
<thead>
<tr>
<th>Configuration parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common</td>
<td>MailNotification</td>
</tr>
<tr>
<td>Common</td>
<td>MailNotification</td>
</tr>
<tr>
<td>Common</td>
<td>MailNotification</td>
</tr>
<tr>
<td>Common</td>
<td>MailNotification</td>
</tr>
</tbody>
</table>
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 policies 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 user 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 on page 55
- Using a password policy on page 55
- Using a password policy on page 55
- Editing password policies on page 57
- Custom scripts for password requirements on page 61
- Deny list for passwords on page 63
- Checking a password on page 64
- Testing generation of a password on page 64
- Password expiry on page 64
- Checking authentication on page 65
- Displaying locked employees and system users on page 65
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 defined the settings for the system user passwords (DialogUser.Password and Person.DialogUserPassword) as well as the access code for a one off 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 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 password policy does not violate the 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 a password policy

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 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.

**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 Designer.

- If you create new custom tables with password columns, assign the VI.Common.Customizer.PwdPolicyColumnEntityLogic customizer to the table definition in Designer.

For detailed 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 Designer, select the **Base data | Security settings | Password policies** category.
2. Select the password policy in the result list.
3. Select **Assign objects**.
4. Select the assignment you want to change in **Assignments**.
5. Select the new password policy to apply from the **Password Policies** menu.
6. Save the changes.

**To reassign a password policy**

1. In Designer, select **Base data | Security settings | Password policies**.
2. Select the password policy in the result list.
3. Click **Add** in the **Assignments** section and enter the following data.

**Table 25: Assigning a Password Policy**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password column</td>
<td>The password column's identifier.</td>
</tr>
<tr>
<td>Apply to</td>
<td>Application scope of the password policy.</td>
</tr>
</tbody>
</table>

*To specify an application scope*

- a. Click the ... button beside the input field.
- b. Select the table which contains the password column under **Table**.
- c. Select the specific base objects under **Apply to**.
- d. Click **OK**.

4. Save the changes.

**Editing password policies**

*To edit a password policy*

1. In Designer, select the **Base data | Security settings | Password policies** category.
2. Select the password policy in the List Editor.
   - OR -  
     Create a new password policy from **Object | New**.
3. Edit the password policy's master data.
4. Save the changes.

**Detailed information about this topic**

- General master data for a password policy on page 57
- Policy settings on page 58
- Character classes for passwords on page 59
- Custom scripts for password requirements on page 61

**General master data for a password policy**

Enter the following master data for a password policy.
Table 26: Master data for a password policy

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display name</td>
<td>Password policy name. Translate the given text using the button.</td>
</tr>
<tr>
<td>Description</td>
<td>Spare text box for additional explanation. Translate the given text using the button.</td>
</tr>
<tr>
<td>Error Message</td>
<td>Custom error message outputted if the policy is not fulfilled. Translate the given text using the button.</td>
</tr>
<tr>
<td>Owner (Application Role)</td>
<td>Application roles whose members can configure the password policies.</td>
</tr>
<tr>
<td>Default policy</td>
<td>Mark as default policy for passwords.</td>
</tr>
</tbody>
</table>

**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.

Related topics

- [Editing password policies](#) on page 57

Policy settings

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

Table 27: Policy settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial password</td>
<td>Initial password for newly created user accounts, employees or system users. If a password is not entered or if a random password is not generated when a user account, employee or a system user is created, the initial password is used.</td>
</tr>
<tr>
<td>Password confirmation</td>
<td>Reconfirm password.</td>
</tr>
<tr>
<td>Minimum Length</td>
<td>Minimum length of the password. Specify the number of characters a password must have.</td>
</tr>
<tr>
<td>Max. length</td>
<td>Maximum length of the password. Specify the number of characters a password can have.</td>
</tr>
<tr>
<td>Max. errors</td>
<td>Maximum number of errors. Set the number of invalid passwords. Only taken into account when logging in to One Identity Manager.</td>
</tr>
</tbody>
</table>
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 reached the number of maximum failed logins, the employee or system user can no longer log in to One Identity Manager.

You can reset the passwords of employees and system users who have been blocked in Password Reset Portal. For more detailed information, see the One Identity Manager Web Portal User Guide.

**Validity period**
- **Meaning**: Maximum age of the password. Enter the length of time a password can be used before it expires.

**Password history**
- **Meaning**: 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.

**Minimum password strength**
- **Meaning**: 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**
- **Meaning**: Specifies whether name properties are permitted or not permitted in the password. If this option is enabled, name properties are not permitted in passwords. The values of the columns for which the Contains name properties for password check option is set are taken into account. Adjust this option in the column definition in Designer. For more detailed information, see the One Identity Manager Configuration Guide.

### Related topics

- Editing password policies on page 57

### Character classes for passwords

Use the Character classes tab to specify which characters are permitted for a password.
## Table 28: Character classes for passwords

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. number letters</td>
<td>Specifies the minimum number of alphabetical characters the password must contain.</td>
</tr>
<tr>
<td>Min. number lowercase</td>
<td>Specifies the minimum number of lowercase letters the password must contain.</td>
</tr>
<tr>
<td>Min. number uppercase</td>
<td>Specifies the minimum number of uppercase letters the password must contain.</td>
</tr>
<tr>
<td>Min. number digits</td>
<td>Specifies the minimum number of digits the password must contain.</td>
</tr>
<tr>
<td>Min. number special characters</td>
<td>Specifies the minimum number of special characters the password must contain.</td>
</tr>
<tr>
<td>Permitted special characters</td>
<td>List of permitted characters.</td>
</tr>
<tr>
<td>Max. identical characters in total</td>
<td>Maximum number of identical characters that can be present in the password in total.</td>
</tr>
<tr>
<td>Max. identical characters in succession</td>
<td>Maximum number of identical character that can be repeated after each other.</td>
</tr>
<tr>
<td>Denied special characters</td>
<td>List of characters, which are not permitted.</td>
</tr>
<tr>
<td>Lowercase not allowed</td>
<td>Specifies whether the password can contain lower case letters. This setting is only applies when passwords are generated.</td>
</tr>
<tr>
<td>Uppercase not allowed</td>
<td>Specifies whether the password can contain upper case letters. This setting is only applies when passwords are generated.</td>
</tr>
<tr>
<td>Digits not allowed</td>
<td>Specifies whether the password can contain digits. This setting is only applies when passwords are generated.</td>
</tr>
<tr>
<td>Special characters not allowed</td>
<td>Specifies whether the password can contain special characters. This setting is only applies when passwords are generated.</td>
</tr>
</tbody>
</table>

### Related topics

- [Editing password policies](#) on page 57
Custom scripts for password requirements

You can implement custom scripts for testing and generating password 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

- Script for checking a password on page 61
- Script for generating a password on page 62

Script for checking a password

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

Syntax for 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 test

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

Example for a script for testing a password

A password cannot start with ? or !. 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. Create your script in the category Script Library in the Designer.
2. Edit the password policy.
   a. In Designer, select the Base data | Security settings | Password policies category.
   b. Select the password policy in List Editor.
   c. Enter the name of the script to be used to check a password in the Check script input field on the Scripts tab.
   d. Save the changes.

Related topics

- Script for generating a password on page 62
- Editing password policies on page 57

Script for generating a password

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 property Entity of the PasswordPolicy class.

Example for a script to generate a password

In random passwords, the script replaces the ? and ! characters, which are not permitted.

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. Create your script in the category **Script Library** in the Designer.
2. Edit the password policy.
   a. In Designer, select the **Base data | Security settings | Password policies** category.
   b. Select the password policy in List Editor.
   c. Enter the name of the script to be used to generate a password in the **Generating script** input field on the **Scripts** tab.
   d. Save the changes.

Related topics
- [Script for checking a password](#) on page 61
- [Editing password policies](#) on page 57

Deny 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. Select **Base Data | Security settings | Restricted passwords** in Designer.
2. Create a new entry with **Object | New** and enter the term to excluded to the list.
3. Save the changes.
Checking a password

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

To test whether a password conforms to the password policy

1. In 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 generation of a password

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 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. 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 7 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 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 36
- Mail templates in One Identity Manager on page 44

**Checking authentication**

The system runs additional validity checks to prevent users from working with established connections, if they were deactivated after they logged in. The check takes place when the next permissions based action on the connection at a fixed interval of 20 minutes.

You can adjust the interval in the configuration parameter Common | Authentication | CheckInterval. Edit the configuration parameter in the Designer.

**Displaying locked employees and system users**

If a user has reached the maximum number of failed logins, the employee or system user can no longer log on to One Identity Manager.

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

Passwords for locked employees and system users can be reset in the Password Reset Portal. For more detailed information, see the One Identity Manager Web Portal User 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.

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

**IMPORTANT:** Consistently book all changes to an object to the change label. It is not possible to add changes of individual properties to the change label at a later date.

In the Database Transporter program, change labels are provided as an export criterion for creating custom configuration packages. When you create a custom configuration package, single object properties are added to the transport package. The properties contain the values given at the time they were added.

You can create and edit change labels in different One Identity Manager tools. The procedure is similar in all tools. Change labels are allocated using different methods depending on the One Identity Manager tool. Changes are normally allocated before or on saving the changes in the database.

**Detailed information about this topic**

- Creating and editing a change label on page 67
- Displaying contents of a change label on page 69
- Booking changes to a change label retrospectively on page 69
- Deleting a change label on page 71

**Creating and editing a change label**

You can create and edit change labels in different One Identity Manager tools. The procedure is similar in all tools. Change labels are allocated using different methods depending on the One Identity Manager tool. Changes are normally allocated before or on saving the changes in the database. The procedure is described below, using Designer as an example.
To create or edit change labels in Designer

1. Select Database | Change management in Designer.
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 🎨.
   - OR -
   Select a change label from the list and open the edit view using 🎨.
4. Enter the following label data.

Table 29: Change label properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change label</td>
<td>Change label name. This name is used to select the change label for allocating the changes or creating a customer transport package.</td>
</tr>
<tr>
<td>Description</td>
<td>Detailed description of the change label</td>
</tr>
<tr>
<td>Parent change label</td>
<td>Specifies a parent label (optional).</td>
</tr>
<tr>
<td>Status</td>
<td>Status of object changes, such as Development, Test, Production.</td>
</tr>
<tr>
<td>Status comments</td>
<td>Additional comments in relation to the status</td>
</tr>
<tr>
<td>Comment</td>
<td>Additional information to enable tracking of changes to a change label</td>
</tr>
<tr>
<td>Label type</td>
<td>Label type for more detailed classification The Change label type is used by default.</td>
</tr>
<tr>
<td>Locked</td>
<td>Indicates if the change label is locked. If a change label is locked, no further changes can be booked to this label.</td>
</tr>
</tbody>
</table>

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

- Displaying contents of a change label on page 69
- Booking changes to a change label retrospectively on page 69
- Deleting a change label on page 71
Displaying contents of a change label

To display the contents of a change label

1. Select Database | Change management in Designer.
2. In the Change management dialog, select the relevant change label in the Change label menu.

The objects that are already assigned to the change label are displayed in the Tagged changes view. 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.

TIP: You will find an overview of change labels in Base data | General | Change label in Designer. However, you do not have any edit options here.

Related topics

- Creating and editing a change label on page 67
- Booking changes to a change label retrospectively on page 69

Booking changes to a change label retrospectively

You can select individual objects and their dependencies 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.

IMPORTANT: Consistently book all changes to an object to the change label. It is not possible to add changes of individual properties to the change label at a later date.
To book objects to a change label retrospectively

1. Select Database | Change management in Designer.
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 text box.
   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. Select the relevant objects in Objects, click 🚐.
   TIP: To select more than one object, use Shift + select or Ctrl + select.
   TIP: You can also use the properties of an object to add that object to a change label.
      - Select the object and open the Properties context menu. You can see which change labels the object belongs to on the Change labels tag.
      Here you can assign a new or an existing change label to the object and its dependent objects.

To remove objects from a change label

1. Select Database | Change management in Designer.
2. In the Change management dialog, use the Change labels menu to select the change label.
3. In Tagged changes, select the objects you want to remove from the change label.
   TIP: To select more than one object, use Shift + select or Ctrl + select.
4. Click the button to remove the objects from the change label.
Related topics

- Creating and editing a change label on page 67
- Displaying contents of a change label on page 69

Deleting a change label

To delete a change label

1. Select Database | Change management in Designer.
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 Abort.
6. To close the Change management dialog, click OK.
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 Manager and in Designer. The following special cases apply:

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

Detailed information about this topic

- Notes on the consistency check on page 72
- Starting a consistency check on page 73
- Logging test results on page 76
- Repairing errors on page 77

Notes on the consistency check

- It is recommended to run consistency checks with an administrative system user.
- To use the Consistency Editor, the user needs the Option to call a consistency check for a database (Common_ConsistencyCheck) program function.
- To use the repair function in the Consistency Editor, the user needs 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 the user does not have any permissions for a certain object, errors may not be identified or repairing errors may fail.
Starting a consistency check

To run a consistency check

1. Start Consistency Editor in Designer or in Manager by selecting **Database | Check data consistency**.
   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 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:
   a. Checking all test objects
      Start this check by pressing ➤.
   
      **NOTE:** To exclude individual test objects from the check, use the **Disable** button to disable these test objects in the list view before the check starts.
   
   b. Checking individual 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 72
- Displaying test objects and the test status on page 74
- Test settings for consistency checks on page 75
Displaying test objects and the 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 3: Consistency Editor with Initialized Data**

![Table 30: List View Information](image)

**Table 30: List View Information**

<table>
<thead>
<tr>
<th>Column</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>Test object name.</td>
</tr>
<tr>
<td>Count</td>
<td>Total number of objects in the database table.</td>
</tr>
<tr>
<td>Verified</td>
<td>Test progress in percent.</td>
</tr>
<tr>
<td>Error</td>
<td>The number of error that occurred during a consistency check.</td>
</tr>
<tr>
<td>Status</td>
<td>Current test status. The status is updated during the consistency check.</td>
</tr>
</tbody>
</table>
Table 31: Meaning of List View Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔍</td>
<td>Test object is currently being test.</td>
</tr>
<tr>
<td>✔</td>
<td>Consistency check was successful for this Test object.</td>
</tr>
<tr>
<td>🟢</td>
<td>Consistency check for this test object is complete but errors occurred.</td>
</tr>
</tbody>
</table>

Table 32: List View Context Menu Items

<table>
<thead>
<tr>
<th>Context Menu Item</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable</td>
<td>Enables selected test object(s) for the period of the consistency check.</td>
</tr>
<tr>
<td>Disable</td>
<td>Disables selected test object(s) for the period of the consistency check.</td>
</tr>
<tr>
<td>Test</td>
<td>Starts execution of the consistency check for the selected test object(s).</td>
</tr>
<tr>
<td>Skip</td>
<td>Skip the test object during the consistency check.</td>
</tr>
</tbody>
</table>

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. Start Consistency Editor in Designer or in Manager by selecting **Database | Check data consistency**.
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 33: Meanings of the icons used for test settings

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌐</td>
<td>Tests are grouped by themes.</td>
</tr>
<tr>
<td>🡫</td>
<td>Tests are grouped by types (database, tables, objects).</td>
</tr>
</tbody>
</table>
Tests are displayed as a list.

Tests are grouped by module association.

Tests with Error severity are displayed.

Tests with Warning severity are displayed.

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)
```

**Table test**

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

**Object test**

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

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

**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**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Icon" /></td>
<td>Shows all error messages.</td>
</tr>
<tr>
<td><img src="#" alt="Icon" /></td>
<td>Only shows errors in the selected objects list view.</td>
</tr>
<tr>
<td><img src="#" alt="Icon" /></td>
<td>A full description of the error is shown in a separate window.</td>
</tr>
<tr>
<td><img src="#" alt="Icon" /></td>
<td>Fixes the error.</td>
</tr>
<tr>
<td>Icon</td>
<td>Meaning</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>📋</td>
<td>Saves the error messages in a log file.</td>
</tr>
<tr>
<td>⚒</td>
<td>Deletes the error messages.</td>
</tr>
</tbody>
</table>

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

**Related topics**
- Repairing errors on page 77

## 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.

   ℹ️ **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.

ℹ️ **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 72
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 status bar indicates that the database needs to be compiled.

Detailed information about this topic

- Compiling a database with the Database Compiler on page 78
- Output of errors and warnings during compilation on page 81

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

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**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web services</td>
<td>The 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.</td>
</tr>
<tr>
<td>Type-safe database model</td>
<td>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.</td>
</tr>
<tr>
<td>Scripts in the Script Library</td>
<td>To compile scripts from the script library, select the following items:</td>
</tr>
</tbody>
</table>

**TIP:** After a schema extension, use this option to compile the database.

**Table 36: Selection for script compilation**

<table>
<thead>
<tr>
<th>Selection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not compile scripts</td>
<td>The scripts in the script library are not compiled.</td>
</tr>
<tr>
<td>Scripts without dependencies</td>
<td>This variant results in script changes only becoming effective when the One Identity Manager tools are restarted.</td>
</tr>
<tr>
<td>Scripts incl. all dependencies</td>
<td>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.</td>
</tr>
<tr>
<td>Templates, tasks, etc.</td>
<td>Specifies whether code snippets, 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.</td>
</tr>
<tr>
<td>Processes</td>
<td>Specify whether processes are compiled. To limit which processes</td>
</tr>
</tbody>
</table>
are to be compiled, use ☰ to show selection options.

### Table 37: Selection for compiling processes

<table>
<thead>
<tr>
<th>Selection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all processes</td>
<td>All processes are compiled.</td>
</tr>
<tr>
<td>modified processes</td>
<td>All processes that have been modified since the last compilation are compiled.</td>
</tr>
<tr>
<td>Selected processes</td>
<td>Select single objects whose processes are to be compiled.</td>
</tr>
</tbody>
</table>

**To select single objects**

a. Click the [...] button.

b. Choose between compiling modified processes, all processes or selected custom processes. You can limit the preselection more.

c. Click OK.

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. Re-compile 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 4: 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 the 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 for your installed version of One Identity Manager are available or 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 on page 83
- Basics for transporting modifications on page 84
- General notes about transporting changes on page 86
- Creating a transport package with the Database Transporter on page 88
- Importing a transport package with the Database Transporter on page 97
- Displaying contents of a transport package on page 98

Types of transport packages

The One Identity Manager schema is customized 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 38: Transport package

<table>
<thead>
<tr>
<th>Transport Package Type</th>
<th>Description</th>
<th>Tool Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration</td>
<td>Migration packages are provided by for the initial database</td>
<td>Configuration</td>
</tr>
</tbody>
</table>
### Transport Package Type

<table>
<thead>
<tr>
<th>Package Type</th>
<th>Description</th>
<th>Tool Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>package</td>
<td>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.</td>
<td>Wizard</td>
</tr>
<tr>
<td>Hotfix package</td>
<td>Hotfix packages are provided to load individual corrections to the default configuration such as templates, scripts, processes, or files into the database.</td>
<td>Database Transporter Software Loader</td>
</tr>
<tr>
<td>Custom configuration package</td>
<td>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.</td>
<td>Database Transporter</td>
</tr>
</tbody>
</table>

**NOTE:** If a hotfix package only contains changed files, load these files into the database using the Software Loader file.

**NOTE:** If more custom configuration adjustments are made to a One Identity Manager database, then create a custom configuration package and import this transport package in the target database with the Database Transporter. There is no support for merging a hotfix package with a custom configuration package into one transport package.

### Related topics

- Basics for transporting modifications on page 84
- Creating a transport package with the Database Transporter on page 88
- Importing a transport package with the Database Transporter on page 97

### 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
  No object was found in the destination database using the primary key or alternative key, therefore 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 executed if the entire system configuration is transported.

Related topics

- General notes about transporting changes on page 86
- Creating a transport package with the Database Transporter on page 88
- Importing a transport package with the Database Transporter on page 97
General notes about 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.

Notes about 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.

Notes about 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.

- 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 Notes about importing the system configuration on page 96.
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 87
- Displaying transport history on page 87
- Creating a transport package with the Database Transporter on page 88
- Importing a transport package with the Database Transporter on page 97
- Displaying contents of a transport package on page 98

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 might want to block processing:

- 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 Designer or 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 Designer and select Help | Transport history.

Creating a transport package with the Database Transporter

To create a transport package

1. In Change & Extend, select Transport custom modifications. This starts the Database Transporter program.
2. Select Create a transport file on the start page.
3. Enter the connection credentials for the One Identity Manager database on the Select database connection page.
4. 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. To create a log file for the export, enable the Create a log file for data export option. The log file is saved in the output directory of the transport file.
5. Enter a description of the transport data on the Show and define transport parameters page.
6. Select the export criteria on the Define transport data page.

NOTE: You can combine multiple export criteria.

Table 39: Export criteria

<table>
<thead>
<tr>
<th>Export criterion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run SQL statements before the data import</td>
<td>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 a transport package on page 90.</td>
</tr>
<tr>
<td>Transporting Favorite Objects</td>
<td>In an initial selection, all modified processes, scripts, reports and mail templates for a specific timeframe are offered. For more information, see Exporting favorite objects on page 90.</td>
</tr>
<tr>
<td>Transport by Change</td>
<td>Transport the changes to objects or object attributes that</td>
</tr>
</tbody>
</table>
### Export criterion | Description
---|---
Label | are summarized in a change label.  
For more information, see Exporting change labels on page 91.
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 92.
Transporting Schema Extensions | Transport custom schema extensions, such as tables, columns, database procedures, features, triggers, views and indexes.  
For more information, see Transporting schema extensions on page 93.
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 94.
Transporting System Configuration | Transport the entire system configuration.  
For more information, see Transporting the system configuration on page 95 and Notes about importing the system configuration on page 96.
Transporting System Files | Transport single files. These files are distributed with the automatic software update.  
For more information, see Exporting system files on page 95.
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 a transport package on page 90.

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

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

**Related topics**
- General notes about transporting changes on page 86
- Importing a transport package with the Database Transporter on page 97
Integrating SQL statements in a transport package

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 an SQL statement may be required for filling initial data in the new columns.

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

To run SQL statements within a transport package

1. In Database Transporter, select the export criterion for running SQL statements. The following export criteria are available:
   - Run SQL statements before the 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 statements directly.
      - OR -
      Using the interface, load a `.sql` file containing the statements.
   b. Use the **button to save to a file.

Related topics

- General notes about transporting changes on page 86
- Creating a transport package with the Database Transporter on page 88

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 Database Transporter, select the **Transport of favorite objects** export criterion.
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 **Transport objects** area lists all selected objects and their dependencies.

**Related topics**

- General notes about transporting changes on page 86
- Creating a transport package with the Database Transporter on page 88

**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 Database Transporter, select the **Transport by change label** export criterion.
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 40: Additional transport settings**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close change label after export</td>
<td>The change label is closed after the transport. No more changes can be booked to this change label.</td>
</tr>
</tbody>
</table>
## Setting

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy dependent objects to the transport package</td>
<td>Objects that are dependent on the selected object and do not have a change label are also copied to the transport.</td>
</tr>
<tr>
<td>Also display closed change labels</td>
<td>Change labels that are already closed are also offered for selection.</td>
</tr>
</tbody>
</table>

## Related topics

- General notes about transporting changes on page 86
- Creating a transport package with the Database Transporter on page 88
- Working with change labels on page 67

## 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 Database Transporter, select the **Transport by change information** export criterion.
2. Specify which changes you want to transport.

### Table 41: user list

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>by me</td>
<td>Only the changes by the logged-in user are added.</td>
</tr>
<tr>
<td>by all users</td>
<td>Changes are added from all users.</td>
</tr>
<tr>
<td>by selected users</td>
<td>Changes are added from selected users.</td>
</tr>
</tbody>
</table>

**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 Timeframe are available.

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

Table 42: Table selection

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire system</td>
<td>Changes are added from all tables.</td>
</tr>
<tr>
<td>System data</td>
<td>Changes are added from the tables of the system data part.</td>
</tr>
<tr>
<td>User data</td>
<td>Changes are added from the tables of the user data part.</td>
</tr>
<tr>
<td>Selected tables</td>
<td>Changes are added from specific tables.</td>
</tr>
</tbody>
</table>

**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 notes about transporting changes on page 86
- Creating a transport package with the Database Transporter on page 88

**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. This customer prefix must be given for the main database. 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, select the Transport schema extensions export criterion.

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

Related topics

- General notes about transporting changes on page 86
- Creating a transport package with the Database Transporter on page 88

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 Designer. For more information about customizing table definitions, see the One Identity Manager Configuration Guide.

To transport single objects and their dependencies

1. In Database Transporter, select the Transport of selected objects and dependencies export criterion.
2. Click the Select button to select the single objects for the transport.
   a. In the Tables area, select the database table from which you want to copy objects to the custom configuration package.
   b. The Dependencies area 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 area 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 **Transport objects** area lists all selected objects and their dependencies.

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

**Related topics**

- General notes about transporting changes [on page 86](#)
- Creating a transport package with the Database Transporter [on page 88](#)

**Exporting system files**

Use this transport method to transport individual files. These files are distributed with the automatic software update.

*To transport new or modified One Identity Manager files*

1. In the Database Transporter, select the **Transport system files** export criterion.
2. Click **Select** and specify the files to transport.

**Related topics**

- General notes about transporting changes [on page 86](#)
- Creating a transport package with the Database Transporter [on page 88](#)

**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 notes about transporting changes on page 86
- Exporting the system configuration on page 96
- Notes about importing the system configuration on page 96
- Transporting schema extensions on page 93

**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 Database Transporter, select the *Transport the system configuration* export criterion.

**Related topics**

- General notes about transporting changes on page 86
- Creating a transport package with the Database Transporter on page 88
- Transporting the system configuration on page 95
- Notes about importing the system configuration on page 96

**Notes about importing the system configuration**

When importing a transport of the system configuration into a target database, you must follow the instructions described under General notes about transporting changes on page 86 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 96
- Protecting individual properties from being overwritten on page 87
- Importing a transport package with the Database Transporter on page 97

**Importing a transport package 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 on the One Identity Manager database.
2. In Change & Extend, select **Transport custom modifications**. This starts the Database Transporter program.
3. Select **Import 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. Select the transport package file browser and click **Open**.
6. Specify your import options on **Select transport file**.

**Table 43: Import options**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a</td>
<td>Enable this option to create a log file for the import. The log file is</td>
</tr>
<tr>
<td>Setting</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>log file for the data import</td>
<td>saved in the output directory of the transport file.</td>
</tr>
<tr>
<td>Import objects singly and ignore errors</td>
<td>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.</td>
</tr>
<tr>
<td>Ignore default data differences</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

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 notes about transporting changes on page 86
- Notes about importing the system configuration on page 96
- Displaying contents of a transport package on page 98

**Displaying contents of a transport package**

**To display the contents of a transport package**

1. Start the Launchpad and log in on the One Identity Manager database.
2. In **Change & Extend**, select **Transport custom modifications**. 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
     a. Click + to select an entry in the transport file and select **Sort in import order** from the context menu.
     b. 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.
     c. 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**
- Importing a transport package with the Database Transporter on page 97
Importing data with 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:

- Different connectors are provided for exchanging data and synchronization between One Identity Manager and other systems. For detailed information, see the relevant One Identity Manager guides.
- You can also use the ScriptComponent process component for regular data imports into One Identity Manager.
- The DataImporterCMD.exe program provides support for imports via the command line.

Detailed information about this topic

- Importing data from a CSV file on page 101
- Importing data from an external database on page 106
- Configuring an import on page 108
- Using an import definition file on page 115
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.
- 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 CSV 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 Data Import and log in to the program.
2. Load the import definition file, if available.

**NOTE:** Leave this field empty if you want to create a new import definition.

3. Select the Import CSV file import procedure.
4. Load the import file and enter any additional data.
5. Specify how the file is structured.
6. Specify how the row structure is set up.
7. Specify a condition for the rows to import.
8. Configure the import.
   a. Assign the data for target tables and target columns of the One Identity Manager database and specify the key columns
   b. Specify the data hierarchy for the import.
   c. Specify options for handling the data.
   d. Define variables that are set on import.
9. Save the import definition file and the import script.
10. Start the import.
11. End the program or start another import.
Loading the CSV file

On the **Load import file** page in Data Import, enter the following data about the import file.

**Table 44: Import file settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>import file</td>
<td>Path to the .csv file containing the data to be imported. You can use the ...</td>
</tr>
<tr>
<td>File encoding</td>
<td>Encoding of the .csv file. Encoding of the character set is determined from the</td>
</tr>
<tr>
<td></td>
<td>character set on your workstation when the import file is loaded. Change the</td>
</tr>
<tr>
<td></td>
<td>setting if the file was created with another character set.</td>
</tr>
<tr>
<td>File culture</td>
<td>File language for the file. The language is required in order to read local</td>
</tr>
<tr>
<td></td>
<td>character formats correctly, for example, dates.</td>
</tr>
<tr>
<td>Time zones</td>
<td>If date and time information is imported, select the time zone of the data.</td>
</tr>
<tr>
<td></td>
<td>The time zone is required for converting the data to UTC.</td>
</tr>
</tbody>
</table>

Structure of the CSV file

On the **File structure** page in Data Import, specify how the file is structured.

**Table 45: File structure**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of</td>
<td>Enter the number of head lines in the .csv file. The header is not</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>lines in header</td>
<td>imported.</td>
</tr>
<tr>
<td>Columns identified by</td>
<td>ID for column limits.</td>
</tr>
<tr>
<td></td>
<td>- Select <strong>Delimiter</strong> if the data is separated by a semi-colon, comma, space, tab, pipe, or other character. Specify the line structure.</td>
</tr>
<tr>
<td></td>
<td>- Select <strong>Fixed width</strong> if all the data in the columns has the same length. Specify the line structure.</td>
</tr>
</tbody>
</table>

Detailed information about this topic

- Specifying the line structure for data with delimiters on page 103
- Specifying the line structure for data with a fixed width on page 105

**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** area displays the line structure according to the selected settings.

**Table 46: line structure**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimiter</td>
<td>Delimiter used to separate the data in the file. You have the following options: <strong>Semicolon</strong>, <strong>Comma</strong>, <strong>Space</strong>, <strong>Tab</strong> and <strong>Pipe</strong>.</td>
</tr>
<tr>
<td></td>
<td>If the data is separated by a different character, select <strong>Other:</strong> and enter the delimiter in the input field next to the menu.</td>
</tr>
<tr>
<td>Text qualifier</td>
<td>Character enclosing the column text. This text is treated as one value on import, even if the text contains the delimiter given as above.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> The delimiters are masked by doubling them up.</td>
</tr>
</tbody>
</table>

Example:

```
Delimiter: Comma (,)
```
<table>
<thead>
<tr>
<th><strong>Property</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Text qualifier:</td>
<td>Quotation mark (&quot;)</td>
</tr>
<tr>
<td>Value in file:</td>
<td>&quot;Smith,Bill&quot;</td>
</tr>
<tr>
<td>Value after import:</td>
<td>Smith,Bill</td>
</tr>
<tr>
<td>Delimiter:</td>
<td>Comma (,)</td>
</tr>
<tr>
<td>Text qualifier:</td>
<td>Not given or other character:</td>
</tr>
<tr>
<td>Value in file:</td>
<td>&quot;Smith,Bill&quot;</td>
</tr>
<tr>
<td>1st value after import:</td>
<td>&quot;Smith&quot;</td>
</tr>
<tr>
<td>2nd value after import:</td>
<td>Bill&quot;</td>
</tr>
</tbody>
</table>

**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: | Bill |
| 3rd value after import: | Bill |

**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 although the rest of the columns remain the same.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Example: | The line  
  John;Smith;Org1|Org2|Org3  
  is converted by suitable settings to the import source  
  John;Smith;Org1  
  John;Smith;Org2  
  John;Smith;Org3 |

**Related topics**
- Structure of the CSV file on page 102
- Specifying the line structure for data with a fixed width on page 105

**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 102
- Specifying the line structure for data with delimiters on page 103

**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 detailed information about scripts in the One Identity Manager, see the One Identity Manager Configuration Guide.
Access using Column Indexing (0...n)

Example:
Do not import the data record if the first column contains the **OLD** value.
Value = $0<>"OLD"

Access using Column Identifier

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 Data Import and log in to the program.
2. Load the import definition file, if available.
   
   🔄 **NOTE:** Leave this field empty if you want to create a new import definition.
3. Select the **Import from database** import procedure.
4. Specify the connection data to the external database.
5. Formulate the source data query.
6. Configure the import.
   a. Assign the data for target tables and target columns of the One Identity Manager database and specify the key columns.
   b. Specify the data hierarchy for the import.
   c. Specify options for handling the data.
   d. Define variables that are set on import.
7. Save the import definition file and the import script.
8. Start the import.
9. End the program or start another import.
Detailed information about this topic

- Selecting an external database on page 107
- Determining the source data on page 108
- Assigning the data to target tables and target columns on page 109
- Specifying the data hierarchy on page 111
- Options for handling records on page 111
- Specifying connection variables on page 113
- Importing the data on page 113
- Using an import definition file on page 115

Selecting an external database

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 area, select the provider of the external database.
   - A list of the various database providers available is shown.

2. In the Connection data area, 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 area. The time zone is required for converting the data to UTC.
Determining the 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>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Tables whose content is imported.</td>
</tr>
</tbody>
</table>
| Columns                | Columns whose content is imported.  
|                        | 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 an 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 the data to target tables and target columns on page 109
- Specifying the data hierarchy on page 111
Assigning the data to target tables and target columns

On the Match target tables and columns page in Data Import, specify how the data is stored in the One Identity Manager database.

To assign target table and target columns
1. In the Target table area, select the target table into which data is imported.
   
   **TIP:** Use the button in the Target table area 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 area, 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 48: Properties for target columns and keys

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use as a key column</td>
<td>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.</td>
</tr>
<tr>
<td>conversion script</td>
<td>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 detailed information about scripts in the One Identity Manager, see the One Identity Manager Configuration Guide.</td>
</tr>
<tr>
<td>target column</td>
<td>Select the target columns to be imported into the data. All columns from the target table are displayed with their data type. Following applies:</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>• Compulsory data is labeled with a blue triangle in front of the data type.</td>
</tr>
<tr>
<td></td>
<td>• Columns without sufficient permissions are displayed in gray.</td>
</tr>
<tr>
<td></td>
<td>• Columns, deactivated by preprocessor condition, are not shown.</td>
</tr>
<tr>
<td>TIP:</td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• Use the Show column captions option to switch between the display name and technical name of the column.</td>
</tr>
</tbody>
</table>

**TIP:** In the assignment wizard, you can use the > button to switch to the next column. The Data preview area contains a preview of the values.

**Related topics**

- Inserting columns with fixed values on page 110

**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 area, click the arrow button beside any column name to open the assignment wizard.
2. Click the button.
3. Enter the value you want in Fixed value.
   - 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 the data to target tables and target columns on page 109
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).

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.

NOTE:

- Sorting the data into a hierarchical structure can consume a great deal of memory. 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.

Related topics

- Determining the source data on page 108

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 49: Options for Handling Records

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert new data set</td>
<td>The data record from the source data does not yet exist in the database.</td>
</tr>
<tr>
<td></td>
<td>If the option is enabled, the data record is inserted in the database.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Adapting existing records</td>
<td>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.</td>
</tr>
<tr>
<td>Delete records that no longer exist</td>
<td>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.</td>
</tr>
<tr>
<td>Limiting the target objects</td>
<td>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.</td>
</tr>
</tbody>
</table>

NOTE: If the Insert new data set option is enabled, source data records that do not fall within the area of relevant database entries 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

![Venn Diagram]

<table>
<thead>
<tr>
<th>Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>All objects in the database.</td>
</tr>
<tr>
<td>B</td>
<td>Database set restricted by condition.</td>
</tr>
<tr>
<td>C</td>
<td>Entry in source data.</td>
</tr>
<tr>
<td>D</td>
<td>All entries in the database and in the source data. Typical action: update all entries in the database.</td>
</tr>
<tr>
<td>E</td>
<td>Entries that are only in the source data but not in the database. Typical</td>
</tr>
<tr>
<td>Case</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>action: add new entry in the database.</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Entries that are in the database but not in the source data. Typical action: clean up entries in the database.</td>
</tr>
<tr>
<td>G</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 executed 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 execute 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 execute the import, use the `DataImport` process task of the `ScriptComponent` process component.
  For more detailed information about creating and editing processes, see the *One Identity Manager Configuration Guide*. 

Start import immediately

To start the import immediately in the Data Import

1. On the Saving the import definition page, enable 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 with Save log as file....

Related topics

- Create an import script on page 114

Create an import script

NOTE: The import script is stored in the One Identity Manager database. To copy import scripts into the database, the user needs 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 text box 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 114
- Working with change labels on page 67
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 132
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 One Identity Manager database using the Software Loader 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 the 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 a One Identity Manager database on page 117
- Exporting files from a One Identity Manager database on page 119
Importing custom files into a One Identity Manager database

**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 on the One Identity Manager database.
2. In **Change & Extend**, select **Import files for software update**. This starts the Software Loader program.
3. Select **Import into database** on the start 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**.
   - 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 50: Meaning of status**

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version unknown</td>
<td>The file belongs to the known files but has not yet been loaded into the database. There is no version information in the database.</td>
</tr>
<tr>
<td>Unknown file</td>
<td>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.</td>
</tr>
<tr>
<td>Version OK</td>
<td>The file version matches the version in the database.</td>
</tr>
<tr>
<td>Version modified</td>
<td>The file version has been modified compared with the version in the database.</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory source</td>
<td>Directory path in installation source.</td>
</tr>
<tr>
<td>Create backup</td>
<td>A copy must be made of the file during the automatic software update.</td>
</tr>
<tr>
<td>No update</td>
<td>The file is not updated by the automatic software update.</td>
</tr>
</tbody>
</table>

9. Click Finish on the last page to end the program.

Related topics
- Exporting files from a One Identity Manager database on page 119
- Editing file settings for the automatic software update on page 119
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. Select the Base data | Installation | One Identity Manager software in the Designer category.
2. Select a file.
3. Edit the following master data.

Table 52: File properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create backup</td>
<td>During the automatic software update, a backup of the existing file is created for files marked with this option.</td>
</tr>
<tr>
<td>No update</td>
<td>The file is not updated by the automatic software update.</td>
</tr>
</tbody>
</table>

Related topics

- Importing custom files into a One Identity Manager database on page 117

Exporting files from a One Identity Manager database

To export files from a One Identity Manager database

1. In Change & Extend, select Import files for software update. This starts the Software Loader program.
2. On the home page, select Export from database.
3. On the Connect to database page, enter the connection data for the One Identity Manager database.
4. Specify which data to export on the **Select files** page.
   a. Specify the destination directory to which to export the data.
      Exportable files are displayed with their status and file size.

      **Table 53: Meaning of status**

      | Status        | Meaning                                                                 |
      |---------------|-------------------------------------------------------------------------|
      | Unknown file  | The file has not yet been exported from the database to the specified directory. |
      | Version OK    | The file version matches the version in the database.                    |
      | Version modified | The file version has been modified compared with the version in the database. |

   b. Mark the files to export.

      **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.

5. 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**.

6. Click **Finish** on the last page to end the program.

**Related topics**

- Importing custom files into a One Identity Manager database on page 117
Appendix: 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 on page 121
- DBCompilerCMD.exe on page 123
- Quantum.MigratorCmd.exe on page 124
- WebDesigner.InstallerCMD.exe on page 126
- VI.WebDesigner.CompilerCmd.exe on page 128
- SoftwareLoaderCMD.exe on page 130
- DBTransporterCMD.exe on page 131
- DataImporterCMD.exe on page 132
- SchemaExtensionCmd.exe on page 134

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 InstallManager.CLI.exe program in the administrative context.

Call 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}]

Example calls

InstallManager.CLI.exe -m install -r c:\source -mod QER ADS SAP LDAP ATT
InstallManager.Cli.exe -m change - r c:\source -d Server\JobServer\ADS

Table 54: Program parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Alternative</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m</td>
<td>--mode</td>
<td>Installation mode. Permitted values are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• install: Install new modules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• change: Update existing modules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• remove: Delete modules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• uninstall: Uninstall complete installation.</td>
</tr>
<tr>
<td>-r</td>
<td>--rootpath</td>
<td>Directory containing the installation sources.</td>
</tr>
<tr>
<td>-i</td>
<td>--installpath</td>
<td>Optional parameter. Directory in which to install.</td>
</tr>
<tr>
<td>-fo</td>
<td>--files-only</td>
<td>Optional parameter. Only file actions are executed. No start menu entries or registry keys are generated and no services are installed.</td>
</tr>
<tr>
<td>-mod</td>
<td>--module</td>
<td>Space-delimited list of module IDs.</td>
</tr>
<tr>
<td>-d</td>
<td>--deploymenttarget</td>
<td>Space delimited list of machine roles.</td>
</tr>
<tr>
<td>-p</td>
<td>--packages</td>
<td>Space-delimited list of packages.</td>
</tr>
<tr>
<td>-l</td>
<td>--logfile</td>
<td>Optional parameter. Path to the log file.</td>
</tr>
<tr>
<td>-fu</td>
<td>--forceupdate</td>
<td>Optional parameter. All data are re-installed.</td>
</tr>
<tr>
<td>-cs</td>
<td>--changeservice</td>
<td>Changes the properties for registration of the service. The following values are expected:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &lt;Service name&gt;: Name of the service to be changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &lt;Properties&gt;: New properties of the service with;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &lt;Name&gt;: Name of the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &lt;Display&gt;: Display name of the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &lt;Description&gt;: Description of the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Name=&lt;New name&gt;;Display=&lt;New display&gt;;Description=&lt;New Description&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You only need to specify the properties that are to be changed.</td>
</tr>
<tr>
<td>-h</td>
<td>--help</td>
<td>Optional parameter. Display program help.</td>
</tr>
</tbody>
</table>
DBCompilerCMD.exe

The DBCompilerCMD.exe program supports compiling a database. You can run the program from the command line.

Call 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] [CompileAp1Projects] [CompileHtmlApps] [FillMultiLanguage]]

Calling 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

Table 55: Program parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Conn</td>
<td>Database connection parameter.</td>
</tr>
<tr>
<td>/Auth</td>
<td>Authentication data. The authentication data depends on the authentication module. For detailed information about the One Identity Manager authentication modules, see the One Identity Manager Authorization and Authentication Guide.</td>
</tr>
<tr>
<td>/LogLevel</td>
<td>Optional parameter. Scope of output to be processed. Permitted values are:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Off</strong>: No log</td>
</tr>
<tr>
<td></td>
<td>- <strong>Fatal</strong>: Logs all critical error messages</td>
</tr>
<tr>
<td></td>
<td>- <strong>Error</strong>: Logs all error messages</td>
</tr>
<tr>
<td></td>
<td>- <strong>Info</strong>: Logs all information</td>
</tr>
<tr>
<td></td>
<td>- <strong>Warn</strong>: Logs all warnings</td>
</tr>
<tr>
<td></td>
<td>- <strong>Debug</strong>: Logs debugger messages. This setting should only be used for testing.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Trace</strong>: Logs highly detailed information. This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome.</td>
</tr>
<tr>
<td>-W</td>
<td>Optional parameter. Wait for the processing of DBQueue Processor jobs to complete before starting compilation.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/Blacklist</td>
<td>Optional parameter. Space-delimited list of compiler modules that must not be compiled. Permitted values are:</td>
</tr>
<tr>
<td></td>
<td>• CompileWebServices: Compile web services</td>
</tr>
<tr>
<td></td>
<td>• CompileTypedWrappers: Compile a type-safe database model</td>
</tr>
<tr>
<td></td>
<td>• CompileDialogScripts: Compile scripts from the script library</td>
</tr>
<tr>
<td></td>
<td>• CompileScripts: Compile templates, formatting scripts and task definitions</td>
</tr>
<tr>
<td></td>
<td>• CompileJobChains: Compile processes</td>
</tr>
<tr>
<td></td>
<td>• CompileWebProjects: Compile web projects</td>
</tr>
<tr>
<td></td>
<td>• CompileApiProjects: Compile API projects</td>
</tr>
<tr>
<td></td>
<td>• CompileHtmlApps: Compile HTML applications</td>
</tr>
<tr>
<td></td>
<td>• FillMultiLanguage: Extract language-dependent texts</td>
</tr>
<tr>
<td>/?</td>
<td>Display program help.</td>
</tr>
</tbody>
</table>

**Quantum.MigratorCmd.exe**

The Quantum.MigratorCmd.exe program supports the migration of a One Identity Manager database. You can run the program from the command line.

**Call syntax**

```plaintext
quantum.migratorcmd.exe /operation=INSTALL|UPDATE|DUMP|IMPORT /connection="{Connection string}" /system=MSSQL /module={Module IDs}[/+] /destination="{Directory}" [/loglevel="Off|Fatal|Error|Info|Warn|Debug|Trace"] [/password={Password}] [/moduleowner={Module ID}] [/hashsize=<Hash size>] [/clear] [/condition=SQL condition]
```

**Calling example**

```plaintext
quantum.migratorcmd.exe /operation=INSTALL /connection="Data Source=<Database server>;Initial Catalog=<Database>;User ID=<Database user>;Password=<Password>" /system=MSSQL /destination="C:\install" /module="TSB,ATT,CPL,HDS,POL,RMB,RMS,RPS"
```

**Table 56: Program parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Alternative</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/operation</td>
<td>-O</td>
<td>-o</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>INSTALL</strong>: Install new database.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Alternative</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>/connection</td>
<td>-C</td>
<td>-c</td>
</tr>
<tr>
<td>/system</td>
<td>-S</td>
<td>-s</td>
</tr>
<tr>
<td>/module</td>
<td>-M</td>
<td>-m</td>
</tr>
<tr>
<td>/password</td>
<td>-P</td>
<td>-p</td>
</tr>
<tr>
<td>/moduleowner</td>
<td>-W</td>
<td>-w</td>
</tr>
<tr>
<td>/format</td>
<td>-F</td>
<td>-f</td>
</tr>
<tr>
<td>/hashsize</td>
<td></td>
<td>For internal use only.</td>
</tr>
<tr>
<td>/destination</td>
<td>-D</td>
<td>-d</td>
</tr>
<tr>
<td>/loglevel</td>
<td></td>
<td>Optional parameter. Scope of output to be processed. Permitted values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Off</strong>: No log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Fatal</strong>: Logs all critical error messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Error</strong>: Logs all error messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Info</strong>: Logs all information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Warn</strong>: Logs all warnings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Debug</strong>: Logs debugger messages This setting should only be used for testing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Trace</strong>: Logs highly detailed information This setting should only be used for analysis purposes. The log file quickly becomes large and cumbersome.</td>
</tr>
<tr>
<td>/clear</td>
<td></td>
<td>For internal use only.</td>
</tr>
<tr>
<td>@filename</td>
<td></td>
<td>As an alternative to directly issuing commands, you can name a text file containing the commands. Every command is in a separate line. Path names in the file must be relative.</td>
</tr>
<tr>
<td>/?</td>
<td>-h</td>
<td>-help</td>
</tr>
</tbody>
</table>
WebDesigner.InstallerCMD.exe

Using the program WebDesigner.InstallerCMD.exe, you can install and uninstall the Web Portal using the command line console.

**NOTE:** Perform the installation using the command line console as an administrator.

**Call 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] [-s]

**Call syntax for uninstalling**

WebDesigner.InstallerCMD.exe [/prov {Provider}] /conn {Connection string} /authprops {Authentication} /appname {Application name} /site {Site} -R

**Call syntax for uninstalling earlier Web Portal versions (<= Version 6.x)**

WebDesigner.InstallerCMD.exe /appname {Application name} /site {Site} -R

**Table 57: Program parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Conn</td>
<td>Database connection parameter.</td>
</tr>
<tr>
<td>/authprops</td>
<td>Authentication data. The authentication data depends on the authentication module used. For detailed information about the One Identity Manager authentication modules, see the One Identity Manager Authorization and Authentication Guide.</td>
</tr>
<tr>
<td>/appname</td>
<td>Application name.</td>
</tr>
<tr>
<td>/site</td>
<td>Website.</td>
</tr>
<tr>
<td>/sourcedir</td>
<td>Optional parameter. If this parameter is enabled, the installation is performed from the file system. If this parameter is not enabled, the installation is performed from the database (default).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/apppool</td>
<td>Optional parameter. If this parameter is enabled, the installation is performed in the specified application pool. If this parameter is not enabled, a new application pool is installed (default).</td>
</tr>
<tr>
<td>/webproject</td>
<td>Optional parameter. Name of the web project. If this parameter is enabled, the specified web project is installed. If this parameter is not enabled, the web project VI_StandardWeb is installed (default).</td>
</tr>
<tr>
<td>/constauthproj</td>
<td>Name of sub project.</td>
</tr>
<tr>
<td>/constauth</td>
<td>Authentication settings for the sub project.</td>
</tr>
<tr>
<td>/searchserviceurl</td>
<td>Application server for search function availability.</td>
</tr>
<tr>
<td>/applicationtoken</td>
<td>Application token for the Password Reset Portal.</td>
</tr>
<tr>
<td>/updateuser</td>
<td>Optional parameter. User for the update.</td>
</tr>
<tr>
<td>/updateuserdomain</td>
<td>Active Directory domain of the user.</td>
</tr>
<tr>
<td>/updateuserpassword</td>
<td>User password.</td>
</tr>
<tr>
<td>/allowhttp</td>
<td>Optional parameter. If this parameter is enabled, HTTP is permitted. If this parameter is not available, HTTPS is used (default).</td>
</tr>
<tr>
<td>-w</td>
<td>Optional parameter. If this parameter is enabled, Windows authentication is used. If this parameter is not enabled, anonymous authentication is used on IIS (default).</td>
</tr>
<tr>
<td>-f</td>
<td>Optional parameter. If this parameter is enabled, no permissions are allocated for the IIS_USRS user. If this parameter is not enabled, the permissions are allocated for the IIS_USRS user (default).</td>
</tr>
<tr>
<td>-s</td>
<td>Optional parameter. If this parameter is enabled, the installation is performed without any user input. If this parameter is not enabled, user input is required (default).</td>
</tr>
<tr>
<td>-R</td>
<td>Delete the web application.</td>
</tr>
<tr>
<td>/?</td>
<td>Program help.</td>
</tr>
</tbody>
</table>

**Example of an installation with direct connection to an SQL Server database.**

In this example, the following settings are made on the parameters.

- Connection to database on an 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 of an installation with direct connection to an application server

In this example, the following settings are made on the parameters.

- Connection to application
- Installation in the "default website"
- Application name "testviaappserver"
- with Windows authentication as web authentication
- User for the update "JohnDoe" with the domain "MyDomain.lan"


Example of uninstalling the web application with a connection against an application server


Example for the processing of authentication settings for a subproject


VI.WebDesigner.CompilerCmd.exe

With the program VI.WebDesigner.CompilerCmd.exe, you can compile the Web Portal using the command line console.
Call syntax

`VI.WebDesigner.CompilerCmd.exe /conn {Connection string} /dialog {Authentication string} /project {path} [/solution {path}] [/mode {mode}] [-E] [-D] [-R] [/csharpout {folder}]`

**Table 58: Program parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Conn</td>
<td>Database connection parameter.</td>
</tr>
<tr>
<td>/dialog</td>
<td>Authentication data The authentication data depends on the authentication module used. For detailed information about the One Identity Manager authentication modules, see the <em>One Identity Manager Authorization and Authentication Guide</em>.</td>
</tr>
<tr>
<td>/project</td>
<td>Name of the web project.</td>
</tr>
<tr>
<td>/solution</td>
<td>Optional parameter. This parameter specifies the Web Designer solution file to be used. If this parameter is not available, a database project is used.</td>
</tr>
</tbody>
</table>
| /mode       | Optional parameter. 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 parameter. This parameter activates the detailed check. For more information about the detailed check, see the *One Identity Manager Web Designer Reference Guide*. |
| -D          | Optional parameter. This parameter activates the debug compilation.          |
| -R          | Optional parameter. This parameter activates the generation of a stable C# text. This setting prevents use of certain random values. |
| /csharpout  | Optional parameter. This parameter contains the target directory for C# text. |
| {folder}    |                                                                              |
| /help       | Program help.                                                               |

**Example based on 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 based on 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
```

**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.

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.

**Call syntax**

```
SoftwareLoaderCMD.exe /Conn="{Connection string}" /Auth="{Authentication String}" [/Root="{Path}"] [-I] /Files="{files|Targets}"```

**Calling example**

Updating files that are known 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
```

Importing customer-specific 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"
```

**Table 59: Program parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Conn</td>
<td>Database connection parameter.</td>
</tr>
<tr>
<td>/Auth</td>
<td>Authentication data The authentication data depends on the authentication module used. For detailed information about the One Identity Manager authentication modules, see the One Identity Manager Authorization and Authentication Guide.</td>
</tr>
<tr>
<td>/Root</td>
<td>Optional parameter. Directory for the files.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-I</td>
<td>Optional parameter. Only in combination with /Files. If this parameter is not set, only the files that are already known in the QBFMFileRevision table are imported. If this parameter is set, unknown files are also imported into the database and an entry is created in the QBFMFileRevision table.</td>
</tr>
<tr>
<td>/Files</td>
<td>List of files with pipe-delimited (</td>
</tr>
<tr>
<td>-N</td>
<td>Optional parameter. If this parameter is set, all files are updated which are known in the QBFMFileRevision table and which are located in the directory specified under /Root. /Conn, /Auth and /Root are mandatory parameters in this mode. -I and /Files are not taken into account.</td>
</tr>
<tr>
<td>-?</td>
<td>Display program help.</td>
</tr>
</tbody>
</table>

### DBTransporterCMD.exe

Using the DBTransporterCMD.exe program, you can import transport packages into the One Identity Manager database. You can run the program from the command line.

#### Call syntax


#### Calling example

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>"

### Table 60: Program parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Conn</td>
<td>Database connection parameter.</td>
</tr>
<tr>
<td>/Auth</td>
<td>Authentication data. The authentication data depends on the authentication module used. For detailed information about the One Identity Manager authentication modules, see the One Identity Manager Authorization and Authentication Guide.</td>
</tr>
</tbody>
</table>
**Parameter** | **Description**
---|---
/File | Transport file to be imported into the database.
-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.

**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.

**Call syntax**

```bash
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]
```

**Example call for importing a single file**

```bash
/Data=VI.DB.ViSqlFactory, VI.DB
/Conn= "Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>"
/Auth=Module=DialogUserAccountBased
/Definition=C:\Work\Import\Data\Def_DataImporter_Employee.xml
/ImportFile=C:\Work\Import\Data\1_Employees.csv
```

**Example call for importing multiple files**

```bash
/Data=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

Table 61: Program parameters
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Conn</td>
<td>Database connection parameter.</td>
</tr>
<tr>
<td>/Auth</td>
<td>Authentication data The authentication data depends on the authentication module used. For detailed information about the One Identity Manager authentication modules, see the One Identity Manager Authorization and Authentication Guide.</td>
</tr>
<tr>
<td>/Definition</td>
<td>Path to the import definition file. Example: C:\Path\To\Definition.xml</td>
</tr>
<tr>
<td>/ImportFile</td>
<td>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</td>
</tr>
<tr>
<td>/DefinitionPair</td>
<td>Pair of the import definition file and the import file. The files are separated by a pipe character (</td>
</tr>
<tr>
<td>/LogLevel</td>
<td>Optional parameter. Scope of output to be processed. Permitted values are: Off: No logging.</td>
</tr>
</tbody>
</table>

Appendix: Command line programs

One Identity Manager 8.1.1 Operational Guide

133
Parameter | Description
---|---
**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 parameter. File language for the file. The language is required in order to read local character formats correctly, for example, dates. Example: de-DE
-p | Optional parameter. If this parameter is used, the processing progress is shown.
-? | Display program help.

Related topics

- [Importing data from a CSV file](#) on page 101

**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 detailed information, see the [One Identity Manager Configuration Guide](#).

**Call syntax**

```
SchemaExtensionCmd.exe /Conn="{Connection string}" /Auth="{Authentication String}"
[/Definition="{Path to import definition file}"
[-f]
[/LogLevel=Off|Fatal|Error|Info|Warn|Debug|Trace]
```
**Calling example**

SchemaExtensionCmd.exe /Conn="Data Source=<Database server>;Initial Catalog=<Database name>;User ID=<Database user>;Password=<Password>" /Auth=Module=DialogUserAccountBased /Definition=CustomExtensions.xml

**Table 62: Program parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Conn</td>
<td>Database connection parameter.</td>
</tr>
<tr>
<td>/Auth</td>
<td>Authentication data. The authentication data depends on the authentication module used. For detailed information about the One Identity Manager authentication modules, see the <em>One Identity Manager Authorization and Authentication Guide</em>.</td>
</tr>
</tbody>
</table>
| /Definition | Path to the control file (XML file)  
Example:  
C:\Path\To\Definition.xml |
| /LogLevel | Optional parameter. 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 parameter. If this parameter is set, the system does not wait for the processing of DBQueue Processor tasks. This can lead to errors if schema extensions are expected that must previously be generated by the DBQueue Processor. |
| -?        | Display program help. |
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 or other inquiries, visit https://www.oneidentity.com/company/contact-us.aspx or call +1-800-306-9329.

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
authentication
  checking 65

C
change label
  assign 69
  create 67
  delete 71
  edit 67
  transporting 91
change labels 67
compilation
  error message 81
  warning 81
compile
  database 78
configuration buffer 84
consistency check 72
  check process test procedure 73
  permission 72
  program function 72
  repair 73
  start 73
  test objects 74
  test setting 75
  test status 74
custom configuration package
  display content 98
  import 97

data export 20
  export definition 21-23
  report 21
  subscribable report 22
data import 100
  change label 113-114
  configure 108
  connection variables 113
  conversion script 109
  CSV file 101
    column index 105
    column name 105
    culture 102
    delimiter 102-103
    encoding 102
    fixed width 102, 105
    headers 102
    import file 102
    line condition 105
    line structure 103, 105
    masking delimiters 103
    text qualifier 103
  CSV import
    time zone 102
  deleting data 111
  dependencies 111
  external database 106
    columns 108
condition (WHERE clause) 108
connection data 107
order by 108
provider 107
query source data 108
select 107
SQL statement 108
table 108
time zone 107
handling records 111
  condition 111
  deleting data 111
  inserting data 111
  updating data 111
hierarchy 111
import definition file 115
  load 115
  save 115
import script 113-114
inserting data 111
log 113
protocol 114
start 114
starting 113
target column 109
  fixed value 110
  key column 109
target table 109
updating data 111
Data Import 100
database
  compile 78
  consistency check 72
  transport history 87
Database Compiler 78
Database Transporter 88, 97
DataImporterCMD.exe
  parameters 132
DBCompilerCMD.exe
  parameters 123
DBTransporterCMD.exe
  parameters 131
Designer
  change labels 67
  compile 78
E
  employee
    locked 65
F
  file
    application group 119
    backup copy 117, 119
    edit 119
    export 119
    import 117
    transport 95
    version 117
H
  hotfix
    display content 98
I
  InstallManager.CLI.exe
    parameters 121
Mail Template Editor

One Identity Manager
system configuration
  TimeTrace 26
One Identity Manager History Database
database 26

Password policies
  editing 57
Password policy
  assign 55
  character classes 59
  check password 64
  default policy 55, 57
  display name 57
  error message 57
  exclusion list 63
  failed logins 58
  generate password 64
  generation script 61-62
  initial password 58
  name properties 58
  password age 58
  password cycle 58
  password length 58
  password strength 58
  predefined 55
  test script 61
planned operations 14
  display 15
  time of execution 15
process component
  ScriptComponent 113

Object
  apply templates 19
historical data 27
process monitoring 30
  data change
    display 34
    object 34
    process 34
    user 34
  process information
    displaying 32
    object 32
    user 32
  process view 30

simulation mode
  simulation data 11
  start 9-10
  stop 9-10

Software Loader 117
software update
  export files 116
  import files 116
SoftwareLoaderCMD.exe
  parameters 130

system user
  locked 65

Q
Quantum.MigratorCmd.exe
  parameters 124

R
report
  simple 21

S
schedule
  activate 38
  configure 36-38
  execution interval 38
  set up 36-38
  start immediately 37
  table 38
  time of execution 39
  time zone 38
  validity period 38
schema extension
  transporting 93

T
template
  re-apply 19

TimeTrace 25-26
  change history 27
  change time stamp 27
  display 27
  prerequisites 26
  reset change 27
  time range 27
  timeline 27
transfer buffer 84
transport package
  basics 84
  change information 92
  change label 91
  create 88
  custom configuration package 83
  date change 92
  display content 98
  export 88
  export criteria 88
full transport 95-96
hotfix package 83
import 97
migration package 83
object selection 94
object selection (favorites) 90
schema extension 93
SQL statement 90
system configuration 95-96
system file 95
transport history 87
user list 92

V
VI.WebDesigner.CompilerCmd.exe
   parameters 128

W
Web Portal
   compile 128
   installing 126
WebDesigner.InstallerCMD.exe
   parameters 126