



One Identity Manager 9.2

Authorization and Authentication Guide

Copyright 2023 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/trademark-information.aspx. All other trademarks are the property of their respective owners.

Legend

 **WARNING:** A WARNING icon highlights a potential risk of bodily injury or property damage, for which industry-standard safety precautions are advised. This icon is often associated with electrical hazards related to hardware.

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

One Identity Manager Authorization and Authentication Guide
Updated - 29 September 2023, 04:03

For the most recent documents and product information, see [Online product documentation](#).

Contents

About this guide	8
One Identity Manager application roles	9
Application roles overview	10
Application roles for basic functions	11
Application role for the Operations Support Web Portal	13
Application role for Compliance & Security Officers	13
Application role for auditors	14
Application roles for identity audit	14
Application roles for company policies	16
Application roles for attestation	17
Application roles for subscribable reports	18
Application roles for management levels	19
Application roles for business roles	19
Application roles for organizations	21
Application role for application roles	22
Application roles for identity administrators	22
Application roles for the IT Shop	23
Application roles for target systems	24
Application roles for Universal Cloud Interface	26
Application role for Privileged Account Governance	27
Application roles for Application Governance	27
Application roles for custom tasks	28
Implementing the application roles	29
Creating and editing application roles	30
Main data of application roles	31
Assigning identities to application roles	32
Custom extension of application role permissions	33
Creating and editing dynamic roles for application roles	34
Specifying mutually exclusive application roles	35
Assigning subscribable reports to application roles	35
Assigning extended properties to application roles	36

Generating assignment resources for application roles	37
Certification of applications roles	37
Reports about application roles	38
Granting One Identity Manager schema permissions through permissions groups	39
Predefined permissions groups and system users	40
Rules for determining the valid permissions for tables and columns	43
Editing permissions groups	45
Dependencies between permissions groups	46
Permissions group dependencies	47
Copying permissions groups	48
Creating permissions groups	49
Permissions group properties	50
Editing system users	50
Creating system users	51
System users' passwords	52
System user properties	53
Adding system users to permission groups	54
Which identities use the system user?	55
Dynamic system user	55
Permissions for tables and columns	56
Displaying permissions of a permissions group	57
Displaying permissions for tables	57
Editing table permissions	58
Editing column permissions	59
Copying table permissions and column permissions	60
Simulating permissions for system users	62
Displaying permissions for objects	63
Displaying permissions for the current user	64
Assigning role-based permissions groups to an applications	64
Managing permissions to program functions	66
Displaying the current user's program functions	66
Assigning program functions to permissions groups	67
Permissions for running scripts	67
Permissions for running methods	68

Permissions for triggering processes	69
Modifying permissions for running actions in the Launchpad	70
One Identity Manager authentication modules	72
System users	73
Generic single sign-on (role-based)	73
Identity	75
Identity (role-based)	76
Identity (dynamic)	77
User account	78
User account (role-based)	79
User account (manual input/role-based)	80
Account based system user	81
Active Directory user account	82
Active Directory user account (role-based)	83
Active Directory user account (manual input)	84
Active Directory user account (manual input/role-based)	85
Active Directory user account (dynamic)	86
LDAP user account (role-based)	87
LDAP user account (dynamic)	90
HTTP header	93
HTTP header (role-based)	95
OAuth 2.0/OpenID Connect	96
OAuth 2.0/OpenID Connect (role-based)	97
Synchronization authentication module	98
Web agent authentication module	99
Component authentication module	99
Crawler	100
Password reset	100
Password reset (role-based)	102
Decentralized identity	104
Decentralized Identity (role-based)	105
Editing authentication modules	106
Enabling authentication modules	106
Assigning authentication modules to applications	107
Disabling or enabling authentication modules for applications	108

Authentication module properties	109
Initial data for authentication modules	110
Configuration data for system user dynamic authentication	114
Example of a simple system user assignment	115
Example of a system user assignment using a selection criterion	116
Example of a function group assignment	117
Checking authentication	118
OAuth 2.0/OpenID Connect authentication	120
Expiry of the OAuth 2.0/OpenID Connect authentication	121
Creating the OAuth 2.0/OpenID Connect configuration	122
Assigning OAuth 2.0/OpenID Connect configuration to web applications	127
Displaying the configuration of the identity provider and the OAuth 2.0/OpenID Connect applications	128
Specifying enabled and disabled columns for logging in	129
Logging information about OAuth 2.0/OpenID Connect authentication	130
Setting up OAuth 2.0/OpenID Connect authentication for accessing the application server's REST API	131
Setting up OAuth 2.0/OpenID Connect authentication the on application server	131
Authentication module for using OAuth 2.0/OpenID Connect for authentication access to the REST API	132
Authenticating external applications on the application server using OAuth 2.0/OpenID Connect	133
OAuth 2.0/OpenID Connect authentication on the API Server	134
Setting up OAuth 2.0/OpenID Connect authentication on the API Server	134
Authenticating external applications on the API Server using OAuth 2.0/OpenID Connect	135
Multi-factor authentication in One Identity Manager	137
Multi-factor authentication with OneLogin	137
Multi-factor authentication with One Identity Defender	138
Configuring RSTS for multi-factor authentication	139
Configuring authentication with OAuth 2.0/OpenID Connect in the Web Portal	140
Configuring authentication with OAuth 2.0/OpenID Connect	141
Granular permissions for the SQL Server and database	143
Displaying database users	143
Displaying users' access levels	144

Displaying database role and server role permissions	144
Installing One Identity Redistributable Secure Token Server	145
Installing One Identity Redistributable Secure Token Server	145
Installing and uninstalling the One Identity Redistributable Secure Token Server	147
Preventing blind SQL injection	148
Appendix: Program functions for starting the One Identity Manager tools ..	150
Appendix: Minimum access levels of One Identity Manager tools	153
About us	156
Contacting us	156
Technical support resources	156
Index	157

About this guide

The *One Identity Manager Authorization and Authentication Guide* describes the basics and features of One Identity Manager's own roles and permissions model.

This shows you an overview of One Identity Manager's application roles, permissions groups, and system users. The guide explains how to set up and implement application roles. The guide also explains how you grant permissions for the tables and columns of the One Identity Manager schema. In addition, you will find an overview of the various One Identity Manager authentication modules.

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.

Available documentation

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

One Identity Manager application roles

You can use the One Identity Manager role model to control edit permissions for One Identity Manager users. This role model takes into account technical aspects, for example, One Identity Manager tool administrative permissions, as well as functional aspects, which result from One Identity Manager user tasks within the company structure (for example, permissions for approving requests). One Identity Manager makes so-called application roles available.

Application roles have the following aims:

- Program functions, identities, company resources, approval workflows and approval policies are assigned to fixed application roles. Permissions for these application roles must not be defined specifically for the company. This simplifies how you manage permissions.
- Enables audit secure internal administration of One Identity Manager users and their permissions. Permissions can be granted through assignment, request, and approval or by calculation on account of specific properties. The plausibility of the permissions can be tested at any time with the attestation function.
- Users are provided with initial permissions, which they required for carrying out their tasks. This is a way, for example, to create initially required user accounts.

Application roles can be linked to permissions groups whose edit permissions are predefined by One Identity Manager. Controlling permissions

- Access to objects and their properties
- Navigation configuration in administration tools
- Which interface forms and tasks are displayed
- Availability of special program functionality

Users must be role-based to use application roles for logging in to One Identity Manager. Role-based authentications module finds the valid edit permissions from all the user's application roles. This provides the One Identity Manager user with permissions corresponding to their application roles for the One Identity Manager functions when they log onto One Identity Manager tools.

Detailed information about this topic

- [Application roles overview](#) on page 10
- [Implementing the application roles](#) on page 29
- [Creating and editing application roles](#) on page 30

Related topics

- [Granting One Identity Manager schema permissions through permissions groups](#) on page 39
- [One Identity Manager authentication modules](#) on page 72

Application roles overview

One Identity Manager supplies default application roles whose permissions are matched to the different task and functions. Assign identities to default applications who take on individual tasks and functions. You can also create your own application roles for custom defined tasks.

NOTE: Default application roles are defined in One Identity Manager modules and are not available until the modules are installed. You cannot delete default application roles.

Detailed information about this topic

- [Application roles for basic functions](#) on page 11
- [Application role for the Operations Support Web Portal](#) on page 13
- [Application role for Compliance & Security Officers](#) on page 13
- [Application role for auditors](#) on page 14
- [Application roles for identity audit](#) on page 14
- [Application roles for company policies](#) on page 16
- [Application roles for attestation](#) on page 17
- [Application roles for subscribable reports](#) on page 18
- [Application roles for management levels](#) on page 19
- [Application roles for business roles](#) on page 19
- [Application roles for organizations](#) on page 21
- [Application role for application roles](#) on page 22
- [Application roles for identity administrators](#) on page 22
- [Application roles for the IT Shop](#) on page 23
- [Application roles for target systems](#) on page 24
- [Application roles for Universal Cloud Interface](#) on page 26

- [Application role for Privileged Account Governance](#) on page 27
- [Application roles for Application Governance](#) on page 27
- [Application roles for custom tasks](#) on page 28

Application roles for basic functions

NOTE: This application role is available if the Identity Management Base Module is installed.

The following application roles are available to you for the basic functionality in One Identity Manager.

Table 1: Application roles for basic functions

Application role	Description
Administrators	<p>Administrators must be assigned to the Base roles Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Administer application roles for administrators. • Assign identities to administrator application roles. • Add other identities to the Base roles Administrators application role and edit conflicting application roles. • See the main data for the other application roles. • Attest application roles' main data. • Can use Password Reset Portal to set passwords for selected system users.
Everyone (change)	<p>The Base roles Everyone (Change) application role is automatically assigned to every user.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Can edit certain identity main data in the Web Portal. <p>If every user is automatically assigned to a custom permissions group when they log in, then this permissions group can be added to the application role.</p> <p>Members of this application role are determined through a dynamic role.</p>
Everyone (lookup)	<p>The Base roles Everyone (Lookup) application role is automatically assigned to every user.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Obtain read access to objects in the Web Portal.

Application role	Description
	<p>If every user is automatically assigned to a custom permissions group when they log in, then this permissions group can be added to the application role.</p> <p>Members of this application role are determined through a dynamic role.</p>
Managers of identities	<p>The Base roles Identity managers application role is automatically assigned to a user if the user is a manager or supervisor of identities, departments, locations, cost centers, business roles, or IT Shops.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Can edit main data for the objects they are responsible for and assign company resources to them. • Can add new identities to the Web Portal and edit the main data of their identities. • Can add their identities to the IT Shop. • Can view their identities' compliance rule violations in the Web Portal. • Can create delegations for their identities the Web Portal. • Can see and edit their identities' delegations in the Web Portal. <p>Members of this application role are determined through a dynamic role.</p>
Birthright Assignments	<p>The Base roles Birthright assignments application role is used to provide birthrights to identities which are provided to establish their working environment. The application roles are allocated all the resources marked for automatic assignment to all identities. All internal identities are assigned to this application role and obtain the resources. Internal identities are found through a dynamic role.</p>
Self-registered identities	<p>All new external identities that have registered themselves in the Web Portal, are assigned to the Base roles Self-registered identities application role. These identities are determined by a dynamic role.</p>

Related topics

- [Custom extension of application role permissions](#) on page 33
- [Application role for the Operations Support Web Portal](#) on page 13

Application role for the Operations Support Web Portal

NOTE: This application role is only available if the Identity Management Base Module is installed.

The Operations Support Web Portal helps you to manage and use your web applications. For more information, see the *One Identity Manager Operations Support Web Portal User Guide*.

The following application roles are available for the Operations Support Web Portal.

Table 2: Application role for the Operations Support Web Portal

Application role	Description
Operations support	Identities that use the Operations Support Web Portal, must be assigned the Base roles Operations support application role. Members of this application role: <ul style="list-style-type: none">• Monitor handling of Job queue processes.• Monitor handling of the DBQueue.• Create passcodes to enable identities to log in to the Password Reset Portal.
Password help desk	Members of the Basic roles Operations support Password help desk application role can reset passwords for other identities in the Operations Support Web Portal.
Synchronization post-processing	Members of the Basic roles Operations support Synchronization post-processing application role are authorized to manage objects in the Operations Support Web Portal that were identified as pending during synchronization.
System administrators	Members of the Base roles Operations support System administrators application role can start and stop processing of the Job queue and the DBQueue in the Operations Support Web Portal.

Application role for Compliance & Security Officers

NOTE: This application role is available if Attestation Module, Compliance Rules Module, or Company Policies Module is installed.

Compliance and security officers must be assigned to the **Identity & Access Governance | Compliance & Security Officer** application role.

Users with this application role:

- View all compliance relevant information and other analysis in the Web Portal. This includes attestation policies, company policies and policy violations, compliance rules, and rule violations and risk index functions.
- Edit attestation policies.

Application role for auditors

NOTE: This application role is available if Attestation Module, Compliance Rules Module or Company Policies Module is installed.

Auditors are assigned to the **Identity & Access Governance | Auditors** application role.

Users with this application role:

- See the Web Portal all the relevant data for an audit.

Application roles for identity audit

NOTE: This application role is available if the Compliance Rules Module is installed.

The following application roles are available for managing compliance rule:

Table 3: Application roles for identity audit

Application role	Description
Administrators	<p>Administrators must be assigned to the Identity & Access Governance Identity Audit Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Enter base data for setting up company policies.• Create compliance rules and assign rule supervisors to them.• Can start rule checking and view rule violations as required.• Create reports about rule violations.• Enter mitigating controls.• Create and edit risk index functions.• Monitor Identity Audit functions.• Administer application roles for rule supervisors, exception approvers and attestors.

Application role	Description
	<ul style="list-style-type: none"> Set up other application roles as required.
Rule supervisors	<p>Rule supervisors must be assigned to the Identity & Access Governance Identity Audit Rule supervisors application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Are responsible for compliance rule content, for example, an auditor or a auditing department. Edit the compliance rule working copies, which are assigned to the application role. Enable and disable compliance rules. Can start rule checking and view rule violations as required. Assign mitigating controls.
Exception approvers	<p>Administrators must be assigned to the Identity & Access Governance Identity Audit Exception approvers application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Edit rule violations in the Web Portal. Can grant exception approval or revoke it in the Web Portal.
Attestors	<p>Attestors must be assigned to the Identity & Access Governance Identity Audit Attestors application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Attest compliance rules and exception approvals in the Web Portal for which they are responsible. Can view main data for these compliance rules but not edit them. <p>NOTE: This application role is available if the module Attestation Module is installed.</p>
Maintain SAP Functions	<p>Administrators must be assigned to the Identity & Access Governance Identity Audit Maintain SAP functions application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Are responsible for SAP function contents. Edit working copies of function definitions for which they are responsible. Define function instances and variables sets for SAP functions.

Application role	Description
	<ul style="list-style-type: none"> Assign mitigating controls. <p>NOTE: This application role is available if the module SAP R/3 Compliance Add-on Module is installed.</p>

Application roles for company policies

NOTE: This application role is available if the Company Policies Module is installed.

The following application roles are available for managing company policies:

Table 4: Application roles for company policies

Application role	Description
Administrators	<p>Administrators must be assigned to the Identity & Access Governance Company policies Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Enter base data for setting up company policies. Set up policies and assign policy supervisors to them. Can calculation policies and view policy violations if required. Set up reports about policy violations. Enter mitigating controls. Create and edit risk index functions. Administer application roles for policy supervisors, exception approvers and attestors. Set up other application roles as required.
Policy supervisors	<p>Policy supervisors must be assigned to the Identity & Access Governance Company policies Policy supervisors application role or another child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Are responsible for the contents of company policies. Edit working copies of company policies. Enable and disable company policies. Can calculation policies and view policy violations if required. Assign mitigating controls.

Application role	Description
Exception approvers	<p>Exception approvers must be assigned to the Identity & Access Governance Company policies Exception approvers application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Edit policy violations. • Can grant exception approval or revoke it.
Attestors	<p>Attestors must be assigned to the Identity & Access Governance Company policies Attestors application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Attest company policies and exception approvals in the Web Portal for which they are responsible. • Can view the main data for these company policies but not edit them. <p>NOTE: This application role is available if the module Attestation Module is installed.</p>

Application roles for attestation

NOTE: This application role is available if the Attestation Module is installed.

The following application roles are available for managing attestation procedures:

Table 5: Application roles for attestation

Application role	Description
Administrators	<p>Administrators are assigned to the Identity & Access Governance Attestation Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Define attestation procedures and attestation policies. • Create approval policies and approval workflows. • Specify which approval procedure to use to find attestors. • Set up attestation case notifications. • Configure attestation schedules. • Enter mitigating controls. • Create and edit risk index functions.

Application role	Description
	<ul style="list-style-type: none"> • Monitor attestation cases. • Manage application roles for attestation policy owners. • Maintain members of the chief approval team.
Chief approval team	<p>The chief approver must be assigned to the Identity & Access Governance Attestation Chief approval team application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Approve using attestation cases. • Assign attestation cases to other attestors.
Attestors for external users	<p>Attestors for external users must be assigned to the Identity & Access Governance Attestation Attestors for external users application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Attests new, external identities.
Attestation policy owner	<p>Owners of attestation policies must be assigned to a child application role of the Identity & Access Governance Attestation Attestation policy owners application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Are responsible for its content and handle the attestation policies assigned to it. • Assign the attestation procedure, approval policy, and calculation schedule. • Assign approvers, mitigating controls, and compliance frameworks. • Monitor attestation cases and attestation runs.

NOTE: Attestors in charge are determined through approval procedures. Other application roles may be applied here. Application roles for attestors are defined in different module and are available if the Attestation Module is installed.

Application roles for subscribable reports

NOTE: This application role is available if the module Report Subscription Module is installed.

The following application role is available for managing subscribable reports:

Table 6: Application roles for subscribable reports

Application role	Description
Administrators	<p>Administrators must be assigned to the Identity & Access Governance Company policies Report Subscriptions application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Create subscribable reports from existing reports.• Configure report parameters for subscribable reports.• Assign subscribable reports to identities, company structures or IT Shop shelves.• Create custom mail templates for sending subscribed reports by email.

Application roles for management levels

NOTE: This application role is available if the module Identity Management Base Module is installed.

The user must be assigned to the **Identity Management | Management level** application role.

Users with this application role:

- Can view reports and statistics in the Web Portal that are intended for their company's management level.

Application roles for business roles

NOTE: This application role is available if the Business Roles Module is installed.

The following application roles are available for the administration of business roles:

Table 7: Application roles for business roles

Application role	Description
Administrators	<p>Administrators must be assigned to the Identity Management Business roles Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Create and edit business roles.

Application role	Description
	<ul style="list-style-type: none"> Assign company resources to business roles. Attest business roles' main data. Administrate application roles for role approvers, role approvers (IT), and attestors. Set up other application roles as required.
Additional managers	<p>The additional managers must be assigned to the Identity Management Business roles Additional managers application role or to a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Have permission to manage business roles.
Attestors	<p>Attestors must be assigned to the Identity Management Business roles Attestors application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Attest correct assignment of company resource to business roles for which they are responsible. Can view main data for these business roles but not edit them. <p>NOTE: This application role is available if the module Attestation Module is installed.</p>
Role approver	<p>Approvers must be assigned to the Identity Management Business roles Role approvers application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Are approvers for the IT Shop. Approve requests from business roles for which they are responsible.
Role approver (IT)	<p>IT role approvers must be assigned to the Identity Management Business roles Role approvers (IT) application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Are IT role approvers for the IT Shop. Approve requests from business roles for which they are responsible.

Application roles for organizations

NOTE: This application role is available if the Identity Management Base Module is installed.

The following application roles are available for the administration of departments, cost centers and locations:

Table 8: Application roles for organizations

Application role	Description
Administrators	<p>Administrators must be assigned to the Identity Management Organizations Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Set up and edit departments, cost centers, and locations.• Assign company resources to departments, cost centers, and locations.• Attest the main data of departments, cost centers, and locations.• Administrate application roles for role approvers, role approvers (IT), and attestors.• Set up other application roles as required.
Additional managers	<p>The additional managers must be assigned to the Identity Management Organizations Additional managers application role or to a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Have permission to manage departments, cost centers and locations.
Attestors	<p>Attestors must be assigned to the Identity Management Organizations Attestors application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Attest correct assignment of company resources to departments, cost centers, and locations for which they are responsible.• Can view main data for departments, cost centers, and locations but cannot edit them. <p>NOTE: This application role is available if the module Attestation Module is installed.</p>
Role approver	<p>Role approvers must be assigned to the Identity Management Organizations Role approvers application role or a child application role.</p>

Application role	Description
	<p>Users with this application role:</p> <ul style="list-style-type: none"> • Are approvers for the IT Shop. • Approve request from departments, cost centers, and locations for which they are responsible.
Role approver (IT)	<p>IT role approvers must be assigned to the Identity Management Organizations Role approvers (IT) application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Are IT role approvers for the IT Shop. • Approve request from departments, cost centers, and locations for which they are responsible.

Application role for application roles

NOTE: This application role is available if the module Identity Management Base Module is installed.

The following application role is available for application role administration.

Table 9: Application roles for organizations

Application role	Description
Additional managers	<p>The additional managers must be assigned to the Identity Management Application roles Additional managers application role or to a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Have permission to manage application roles.

Application roles for identity administrators

NOTE: This application role is available if the module Identity Management Base Module is installed.

The following application role is available for identity administration:

Table 10: Application roles for identities

Application role	Description
Administrators	<p>Identity administrators must be assigned to the Identity Management Identities Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Can edit any identity's main data• Assign managers to identities.• Can assign company resources to identities.• Check and authorize identity main data.• Create and edit risk index functions.• Edit password policies of identities' passwords.• Delete identity's security keys (WebAuthn)• Can see everyone's requests, attestations, and delegations and edit delegations in the Web Portal.

Application roles for the IT Shop

NOTE: This application role is available if the Identity Management Base Module is installed.

The following application roles are available for the IT Shop administration:

Table 11: Application roles for the IT Shop

Application role	Description
Administrators	<p>Administrators must be assigned to the Request & Fulfillment IT Shop Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Create the IT Shop structure with shops, shelves, customers, templates, and service catalog.• Create approval policies and approval workflows.• Specify which approval procedure to use to find attestors.• Create products and service items.• Set up request notifications.• Monitor request procedures.• Administrate application roles for product owners and attestors.

Application role	Description
	<ul style="list-style-type: none"> • Maintain members of the chief approval team. • Set up other application roles as required. • Create extended properties for company resources of any type. • Edit the resources and assign them to IT Shop structures. • Assign system entitlements to IT Shop structures.
Product owners	<p>Product owners must be assigned to the Request & Fulfillment IT Shop Product owners application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Approve through requests. • Edit service items and service categories under their management.
Attestors	<p>Attestors must be assigned to the Request & Fulfillment IT Shop Attestors application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Attest correct assignment of company resource to IT Shop structures for which they are responsible. • Attest objects that have service items assigned to them. • Can view main data for these IT Shop structures but not edit them. <p>NOTE: This application role is available if the Attestation Module is installed.</p>
Chief approval team	<p>Chief approvers must be assigned to the Request & Fulfillment IT Shop Chief approval team application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Approve through requests. • Assign requests to other approvers.

NOTE: The approvers responsible are determined through approval procedures. Other application roles may be applied here. Application roles for approvers are defined in different modules and are available there.

Application roles for target systems

NOTE: Application roles are dependent on the target system and are contained in One Identity Manager modules. Application roles are not available until the modules are

installed.

The following application roles are available for target system administration:

Table 12: Application roles for target systems

Application role	Tasks
Administrators	<p>Target system administrators must be assigned to the Target systems Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Administer application roles for individual target system types.• Specify the target system manager.• Set up other application roles for target system managers if required.• Specify which application roles for target system managers are mutually exclusive.• Authorize other identities to be target system administrators.• Do not assume any administrative tasks within the target system.
Target system managers	<p>Target system managers must be assigned to the Target systems <target system> application role or a child application role.</p> <p>NOTE: There is at least one application role per target system for target system managers. This application role is available if the target system module is installed.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Assume administrative tasks for the target system.• Create, change, or delete target system objects.• Edit password policies for the target system.• Prepare system entitlements to add to the IT Shop.• Can add identities that do not have the Primary identity identity type.• Configure synchronization in the Synchronization Editor and define the mapping for comparing target systems and One Identity Manager.• Edit the synchronization's target system types and outstanding objects.• Authorize other identities within their area of responsibility as target system managers and create child

Application role	Tasks
	application roles if required.
Target system managers for Unified Namespace	<p>Target system managers must be assigned to the Target systems Unified Namespace application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Obtain view of the objects in the connected target systems across all target systems. • Can create reports across all target systems. <p>If the users are also target system managers of the basic underlying target systems, you can manage these target systems through the Unified Namespace.</p>

Application roles for Universal Cloud Interface

NOTE: Application roles are available if the Universal Cloud Interface Module is installed. The following application roles are available for managing cloud systems.

Table 13: Application roles for Universal Cloud Interface

Application role	Tasks
Cloud administrators	<p>Cloud administrators must be assigned to the Universal Cloud Interface Administrators application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> • Manage application roles for the Universal Cloud Interface. • Set up other application roles as required. • Configure synchronization in the Synchronization Editor and define the mapping for comparing cloud applications and One Identity Manager. • Edit cloud application in the Manager. • Edit pending, manual provisioning processes in the Web Portal and obtain statistics. • Obtain information about the cloud objects in the Web Portal and the Manager.

Application role	Tasks
Cloud operators	<p>The cloud operators must be assigned to the Universal Cloud Interface Operators application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Edit pending, manual provisioning processes in the Web Portal and obtain statistics.
Cloud auditors	<p>The cloud auditors must be assigned to the Universal Cloud Interface Auditors application role or a child application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Can view manual provisioning processes in the Web Portal and obtain statistics.

Application role for Privileged Account Governance

NOTE: This application role is available if the module Privileged Account Governance Module is installed.

The following application role is available for managing asset and account owners

Table 14: Application role for Privileged Account Governance

Application role	Description
Asset and account owners	<p>Owners of privileged objects, such as PAM assets, PAM asset accounts, PAM directory accounts, PAM asset groups, and PAM account groups must be assigned to an application role under the Privileged Account Governance Asset and account owners application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none"> Make decisions about requesting access requests for privileged objects. Attest the possible user access to these privileged objects.

Application roles for Application Governance

NOTE: This application role is available if the module Application Governance Module is installed.

Table 15: Application roles for Application Governance

Application role	Tasks
Administrators	<p>Administrators must be assigned to the Application Governance Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Create new applications in the Web Portal.• Manage all applications in the Web Portal.
Owner	<p>The owners of applications must be assigned to the Application Governance Owners application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Can edit applications in the Web Portal for which you are responsible.
Approver	<p>Approvers must be assigned to the Application Governance Approvers application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Approve requests for application products.

Application roles for custom tasks

NOTE: This application role is available if the Identity Management Base Module is installed.

The following application roles are available for customer features and tasks.

Table 16: Application roles for custom tasks

Application role	Description
Administrators	<p>Administrators must be assigned to the Custom Administrators application role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Administrate custom application roles.• Set up other application roles for managers if required.
Manager/supervisor	<p>Managers must be assigned to the Custom Managers application role or a child role.</p> <p>Users with this application role:</p> <ul style="list-style-type: none">• Add custom task in One Identity Manager.• Configure and start synchronization in the

Application role	Description
	<p>Synchronization Editor.</p> <ul style="list-style-type: none"> Edit the synchronization's target system types as well as outstanding objects in the Manager. <p>You can use these application roles, for example, to guarantee One Identity Manager user permissions on custom tables or columns. All application roles that you define here must obtain their permissions through custom permissions groups.</p>

Implementing the application roles

IMPORTANT: To use application roles you must add one identity to the **Base roles | Administrators** application role. This identity is the authorized to assigned administrative One Identity Manager application roles to other identities.

Run this task once.

To initially add an identity to the Base roles | Administrators application role.

1. Log into the Manager as a non role-based administrative user.
2. Select the **Identities > Identities** category.
3. Select the identity to be assigned to the **Base role | Administrators** application role.
4. Select the **Authorize as One Identity Manager administrator** task.

The One Identity Manager user with the **Base roles | Administrators** application role can now add more identities to application roles and edit the application role main data.

NOTE: Once you update the view in the Manager, the **Authorize as One Identity Manager administrator** task is no longer displayed in the task view. That means that the task can only be run when there are no other identities assigned to this application role.

After you have been working with One Identity Manager for a while, it is possible that no more identities are assigned to the **Base roles | Administrators** application role. In this case, proceed as described above in order to reassign an identity to this application role.

Related topics

- [Assigning identities to application roles](#) on page 32
- [Creating and editing application roles](#) on page 30

Creating and editing application roles

To set up your first application roles you need to add an identity to the application role **Base roles | Administrators**. This identity is authorized to add more identities to different administration application roles. For more information, see [Implementing the application roles](#) on page 29.


Administrators can edit child application roles, set up more application roles and assigned identities.

NOTE: To edit the application role, log on to the Manager using a role-based authentication module.

To edit an application role

1. In the Manager in the **One Identity Manager Administration** category, select the Application role.
2. Select the **Change main data** task.
3. Edit the application role's main data.
4. Save the changes.

To create a new application role

1. In the Manager in the **One Identity Manager Administration** category, select the application role under which you want to create a new application role.
2. Click  in the result list.
3. Enter the application role main data.
4. Save the changes.


NOTE: You cannot delete default application roles.

Related topics

- [Main data of application roles](#) on page 31
- [Assigning identities to application roles](#) on page 32
- [Custom extension of application role permissions](#) on page 33
- [Creating and editing dynamic roles for application roles](#) on page 34
- [One Identity Manager authentication modules](#) on page 72

Main data of application roles

Table 17: Application role properties

Property	Meaning
Application role	Application role name.
Internal name	Empty text field for a internal company identifier
Full name	Full name of application role. Is made up automatically from the application role name and the parent application role.
Parent application role	Application role to which the application role being edited is subordinate.
Department, location, cost center	Additional information for the application role definition. These input fields are only used for information. They do not indicate for which department, cost center or location the application roles are responsible.
Manager	Manager responsible for the application role.
Deputy manager	Deputy manager for the application role.
Additional manager	<p>Application role for a group of managers and deputies who manage this application role.</p> <p>To create a new application role, click . Enter the application role name and assign a parent application role.</p>
Permissions group	<p>Permissions group for determining permissions for role-based login. The application role is given the permissions of the associated permissions group. If no permissions group assigned, the application role is obtains the permissions from the parent application role.</p> <p>Administrators can assign the rest of the application roles to custom defined permissions groups.</p> <p>NOTE: Permissions groups for default administrator application roles for cannot be edited.</p>
Description	Text field for additional explanation.
Comment	Text field for additional explanation.
Certification status	<p>Status of the application role's certification. The following values can be selected.</p> <ul style="list-style-type: none">• New: The application role was newly created in the One Identity Manager database.• Certified: The main data of the application role is approved by a manager.

Property	Meaning
	<ul style="list-style-type: none"> • Denied: The application role main data was not approved by a manager. <p>The certification status can be set depending on the result of regular attestations.</p>
Block inheritance	<p>Specifies whether inheritance for this application role can be discontinued. Set this option to prevent company resources being inherited by child application roles.</p> <p>NOTE: Inheritance of application roles can only be discontinued if they are custom application roles.</p>
Dynamic roles not allowed	Specifies whether a dynamic role can be created for the application role.
Spare field no. 01 ... Spare field no. 10	Additional company-specific information. Use the Designer to customize display names, formats, and templates for the input fields.

Related topics

- [Application role for application roles](#) on page 22
- [Custom extension of application role permissions](#) on page 33
- [Creating and editing dynamic roles for application roles](#) on page 34
- [Certification of applications roles](#) on page 37

Assigning identities to application roles

Assigned identities obtain all the permissions of the permission group to which the application role (or a parent application role) is assigned. In addition, identities obtain the company resources assigned to the application role.

If there are no identities directly assigned to an application role, the identities of the parent application role inherit the permissions.

NOTE: The application roles for **Base roles | Everyone (Change)**, **Base roles | Everyone (Lookup)**, **Base roles | Identity Managers**, and **Base roles | Birthright Assignments** are automatically assigned to identities. Do not make any manually assignments to these application roles.


To assign identities to an application role

1. In the Manager, select an application role in the **One Identity Manager Administration** category.
2. Select the **Assign identities** task.

3. In the **Add assignments** pane, add identities.

TIP: In the **Remove assignments** pane, you can remove assigned identities.

To remove an assignment

- Select the identity and double-click .
4. Save the changes.

Related topics

- [Creating and editing dynamic roles for application roles](#) on page 34

Custom extension of application role permissions

For role-based login, application roles must link to a permissions group in which permissions for One Identity Manager are defined. The application role is given the permissions of the associated permissions group. If no permissions group assigned, the application role obtains the permissions from the parent application role.

Some of the default application roles are already assigned permissions groups. These permissions groups have the permissions for the tables and columns and are equipped with menu items, forms, tasks, and program functions, which allow the application data to be edited in the Manager and in the Web Portal.

You can assign customized permissions groups to application roles so that the permissions for application roles meet your company requirements. You need to ensure that your custom permissions groups contain all the write permissions of the default permissions groups for these application roles. This allows users with these application roles to use all default One Identity Manager functionality.

NOTE: You can simplify grouping of permissions by using hierarchical linking of permissions groups. Permissions from hierarchical permissions groups are inherited from top to bottom. That means that a permissions group contains all the permissions belonging parent permissions groups.

Proceed as follows:

1. In the Designer, create a new permissions group .
NOTE: Set the **Only use for role-based authentication** option for the permissions group.
2. In the Designer, make the new permissions group dependent on the default permissions group of the application role. Assign the default permissions group as a parent permissions group. This means the newly defined permissions group inherits the properties of the default permissions group.
3. In the Designer, grant additional edit permissions for menu items, forms, tables, or columns.
4. In the Manager, assign the new permissions group to the application role.

A user who logs in to the Manager or to the Web Portal with an application role changed in this way receives – in addition to the default privileges of this application role – the custom permissions.

Related topics

- [Main data of application roles](#) on page 31
- [Granting One Identity Manager schema permissions through permissions groups](#) on page 39

Creating and editing dynamic roles for application roles

Use this task to assign identities to an application role through dynamic roles. For more information about using dynamic roles, see the *One Identity Manager Identity Management Base Module Administration Guide*.

NOTE: The task **Create dynamic role** is only available for application roles that do not have the option **Dynamic roles not allowed** set.

To create a dynamic role for the application role

1. In the Manager in the **One Identity Manager Administration** category, select the application role.
2. Select the **Create dynamic role** task.
3. Enter the required main data. The following applies to dynamic roles for application roles:
 - **Object class:** Select **Identity**.
 - **Application role:** This data is preset with the selected application role. If these objects fulfill the dynamic role conditions, they become members in the application role.
 - **Dynamic role:** The dynamic role name is made up of the object class and the full name of the application role by default.
4. Save the changes.

To edit a dynamic role

1. In the Manager in the **One Identity Manager Administration** category, select the application role.
2. Select the **Application role overview** task.
3. In the overview form, click the dynamic role name in the **Dynamic roles** form element.
4. Select the **Change main data** task.

5. Edit the dynamic role.
6. Save the changes.

Related topics

- [Main data of application roles](#) on page 31

Specifying mutually exclusive application roles

It is possible that identities cannot own certain system roles at the same time. Thus, for example, exception approvers for rule violations may not be rule supervisors at the same time. To implement this behavior, you can specify mutually exclusive application roles. Then you cannot assign these application roles to the same identity anymore.

NOTE: Only system roles, which are defined directly as conflicting application roles, cannot be assigned to the same identity. Definitions made on parent or child application roles do not effect the assignment.

To configure inheritance exclusion

- In the Designer, set the **QER | Structures | ExcludeStructures** configuration parameter and compile the database.

To specify inheritance exclusion for application roles

1. In the Manager in the **One Identity Manager Administration** category, select the application role for which you want to define an inheritance exclusion.
2. Select the **Edit conflicting application roles** task.
3. In the **Add assignments** pane, assign application roles that are mutually exclusive to the selected system role.
- OR -
In the **Remove assignments** pane, remove the application roles that are no longer mutually exclusive.
4. Save the changes.

Assigning subscribable reports to application roles

Use this task to assign subscribable reports to an application role. All identity in this application role can subscribe to reports in the Web Portal. For more information about subscribable report, see the *One Identity Manager Report Subscriptions Administration Guide*.

NOTE:


- This function is only available if the Report Subscription Module is installed.
- The task is only available if a permissions group is assigned to the application role (or a parent application role).
- Subscribable reports cannot be assigned to the **Base roles | Managers of identities**, the **Base roles | Everyone (Lookup)**, or the **Base roles | Everyone (Change)** application role.

To assign subscribable reports to an application role

1. In the Manager, select an application role in the **One Identity Manager Administration** category.
2. Select the **Assign subscribable reports** task.
3. In the **Add assignments** pane, assign reports.

TIP: In the **Remove assignments** pane, you can remove report assignments.

To remove an assignment

- Select the report and double-click .
4. Save the changes.

Assigning extended properties to application roles


Extended properties are meta objects, such as operating codes, cost codes, or cost accounting areas that cannot be mapped directly in One Identity Manager. For more information about using extended properties, see the *One Identity Manager Identity Management Base Module Administration Guide*.

To specify extended properties for an application role

1. In the Manager, in the **One Identity Manager Administration** category, select the Application role.
2. Select the **Assign extended properties** task.
3. In the **Add assignments** pane, assign extended properties.

TIP: In the **Remove assignments** pane, you can remove assigned extended properties.

To remove an assignment

- Select the extended property and double-click .
4. Save the changes.

Generating assignment resources for application roles

It is possible to create assignment resources for individual application roles. This means you can limit assignment resources to individual application roles in the Web Portal. When the assignment resource is requested, it is no longer necessary to select the application role as well. The application role is automatically a part of the assignment request. For more information about assignment requests, see the *One Identity Manager IT Shop Administration Guide*.

To limit an assignment resource to one application role

1. In the Manager in the **One Identity Manager Administration** category, select the Application role.
2. Select the **Create assignment resource** task.

This starts a wizard that takes you through the steps for adding an assignment resource.

Certification of applications roles

| NOTE: This function is only available if the Attestation Module is installed.

The certification status of application roles can be set manually or by regular attestation. To set certification status by attesting, configure the attestation policies accordingly.

To manually change the certification status of an application role

1. In the Manager, edit the application role's main data.
2. In the **Certification status** field, enter the required value.
3. Save the changes.

To change the certification status of application roles by attestation

1. In the Manager, select the **Attestation > Attestation policies** category.
2. In the result list, select the attestation policy whose attestation runs will adjust the certification status.
3. If the certification status is to change to **Certified** when attestation is approved, enable the **Set certification status to "Certified"**.
4. If the certification status is to be changed to **Denied** when attestation is denied, enable **Set certification status to "Denied"**.
5. Save the changes.

One Identity Manager provides default procedures for managers to quickly attest and certify the main data of newly added application roles in the One Identity Manager database. Attestation is performed only for application roles with the **New** certification

status. If the attestation is approved, the certificate status of the attested application role is set to **Certified** and otherwise, to **Denied**.

NOTE: If the attestation was denied, only the certification status changes. Other behavioral changes, for example in the inheritance calculation, are not associated with this and can be implemented on a custom basis.

This function is only available if the Target System Base Module is installed. For more information about certifying new roles and organizations, see the *One Identity Manager Attestation Administration Guide*.

Detailed information about this topic

- [Creating and editing application roles](#) on page 30
- [Main data of application roles](#) on page 31

Reports about application roles

One Identity Manager makes various reports available containing information about the selected base object and its relations to other One Identity Manager database objects. The following reports are available for application roles.

Table 18: Reports about application roles

Report	Description
Overview of all assignments	This report identifies all departments, cost centers, locations, business roles or IT Shop structures in which identities from the selected application role are also members. For more information about analyzing role memberships, see the <i>One Identity Manager Identity Management Base Module Administration Guide</i> .
Show historical memberships	This report lists all members of the selected application role and the length of their membership.

Granting One Identity Manager schema permissions through permissions groups

Permissions for accessing tables and columns of the One Identity Manager schema are themselves mapped in the schema through permissions groups. You can assign permissions groups to system users and to application roles.

Permissions groups are also used to control access to parts of the user interface, such as, menu items, forms, tasks, and program functions. When a user logs in to One Identity Manager tools, all available menus, forms, and methods are loaded depending on the system user's permissions groups, displaying a user interface customized for this system user. For more detailed information about editing the user interface, see the *One Identity Manager Configuration Guide*.

One Identity Manager provides permissions groups and system users with a predefined user interface and special permissions for One Identity Manager schema's tables and columns. These predefined configurations are maintained by the schema installation and cannot be edited apart from a few properties.

Detailed information about this topic

- [Predefined permissions groups and system users](#) on page 40
- [Rules for determining the valid permissions for tables and columns](#) on page 43
- [Editing permissions groups](#) on page 45
- [Editing system users](#) on page 50
- [Permissions for tables and columns](#) on page 56
- [Managing permissions to program functions](#) on page 66
- [Displaying permissions for objects](#) on page 63
- [Displaying permissions for the current user](#) on page 64
- [Assigning role-based permissions groups to an applications](#) on page 64

Related topics

- [One Identity Manager application roles](#) on page 9
- [One Identity Manager authentication modules](#) on page 72

Predefined permissions groups and system users

One Identity Manager provides permissions groups and system users with a predefined user interface and special permissions for One Identity Manager schema's tables and columns. These predefined configurations are maintained by the schema installation and cannot be edited apart from a few properties.

Table 19: Predefined permissions groups

Permissions group	Description
Permissions group QBM_BaseRights	The QBM_BaseRights permissions group defines the base rights that are required for a system user to log in to the One Identity Manager tools. This permissions group is always assigned implicitly.
Permission group VID_Features	The VID_Features permissions group covers all program functions required for starting the One Identity Manager tools. The permissions group covers additional program functions for running special functions in One Identity Manager.
Permission group VID_View	The VI_View permissions group has viewing permissions for all tables and columns that map application data. NOTE: Assign viewing permissions of custom schema extensions to the permissions group.
Permission group VID_Everyone	The VI_Everyone permissions group is assigned to elements of the overview forms that use links to the corresponding menu items. These permissions groups also provide functions for Web Portal users. NOTE: Assign the permissions group to your custom system users such that the overview form is fully displayed to the users.
Permissions groups for One Identity Manager application data	The permissions groups have permissions on the tables and the columns that map application data. These permissions groups are equipped with menu items, forms, tasks, and program functions which allows the application data to be edited with, for example, the Manager.

Permissions group	Description
Permissions groups for One Identity Manager system data	<p>The permissions groups have permissions on the tables and the columns that map the One Identity Manager's system data. These permissions groups are equipped with menu items, forms, tasks, and program functionality which allows the application data to be edited, for example, with Designer editors.</p> <p>The vid permissions group has all edit permissions for the system configuration with the Designer.</p>
Role-based permissions group VI_4_ALLUSER	The VI_4_ALLUSER permissions group provides the base permissions as well as menu items, forms, tasks, and program functions to enable the application data to be edited with the Manager and the Web Portal. This permissions group is always assigned implicitly.
Role-based permissions group VI_4_ADMIN_LOOKUP	<p>The vi_4_ADMIN_LOOKUP permissions group has the viewing permissions for all tables and columns of the application data.</p> <p>NOTE: Assign viewing permissions of custom schema extensions to the permissions group.</p>
Role-based permissions group QER_4_OperationsSupport	The QER_4_OperationsSupport permissions group has special permissions for working with the Operations Support Web Portal. The permissions group is assigned to the OperationsSupportWebPortal application. The permissions of the permissions group apply only in the Operations Support Web Portal.
Role-based permissions groups	Role-based permissions groups have permissions on the tables and the columns that map application data. These permissions groups are equipped with menu items, forms, tasks, and program functionality which allow the application data to be edited with the Manager and the Web Portal. These permissions groups are linked to the One Identity Manager application roles and simplify administration of access permissions in the One Identity Manager role model.

Table 20: Predefined system users

System users	Description
Dynamic system user	<p>Dynamic system users are used for logging into One Identity Manager tools with role-based authentication modules. First, the identity memberships in the One Identity Manager application roles are determined during login. Assignments of permissions groups to One Identity Manager application roles are used to determine which permissions groups apply to the identity. A dynamic system user is determined from these permissions groups that</p>

System users	Description
	will be used for the identity's login.
System user sa	The sa system user is used exclusively by the One Identity Manager Service. This system user is not assigned to a permissions group but has all the permissions, tasks, and program functionality.
System user viadmin	<p>The viadmin system user is the default system user in One Identity Manager. This system user can be used to compile and initialize the One Identity Manager database and for the first user login to the administration tools.</p> <p>IMPORTANT: Do not use the viadmin system user in a live environment. Create your own system user with the appropriate permissions.</p> <p>The system user has all of the specified permissions and the complete user interface. The system user implicitly receives the authorizations and user interface parts of the custom permissions groups. The system user has the permission to set up an identity as a One Identity Manager administrator for the role-based login. The system user is not a member of the application role themselves.</p>
System user Synchronization	The Synchronization system user has the necessary permissions to set up and run target system synchronizations using an application server.
System user viHelpdesk	The viHelpdesk system user has the necessary permissions and the user interface to use the Manager to access One Identity Manager help desk resources.
System user IdentityRegistration	<p>The IdentityRegistration system user has the specified permissions required for self-registration of new users in the Password Reset Portal.</p> <p>NOTE: This system user is available if the Attestation Module is installed.</p>

Related topics

- [Dependencies between permissions groups](#) on page 46
- [Editing permissions groups](#) on page 45
- [Creating system users](#) on page 51
- [Dynamic system user](#) on page 55

Rules for determining the valid permissions for tables and columns

When a system user is used to log into the system, the permissions for the objects that are currently in effect, are determined on the basis of the permissions groups. The following rules are used to determine the resulting permissions:

- Permissions from hierarchical permissions groups are inherited from top to bottom. That means that a permissions group contains all the permissions belonging parent permissions groups.
- The number of objects is determined first for hierarchical permissions groups. Column permissions are decided afterward. In some cases, this results in more permissions than are defined on individual permissions groups.
- A system user receives a permission when at least one of its permissions groups has the permission (directly or inherited).
- The limiting permissions conditions for all the system user's permissions groups are grouped together and used to determine a valid condition for each permission for viewing, editing, inserting, and deleting an object.
- Fixed viewing permissions for the One Identity Manager schema's system data are granted by the system, which are sufficient for logging a system user into the administration tools.
- A system user with read-only permissions only obtains viewing permissions to objects irrespective of any other permissions.
- If permissions are granted on a table for inserting, editing, or deleting, viewing permissions are implicit.
- If permissions are granted on a column for inserting, editing, or deleting, viewing permissions are implicit.
- If permissions are granted for a table, then viewing permissions are implicitly granted on the primary key column of the table.
- If viewing permissions are granted on a primary key column as a minimum, then viewing permissions are implicitly granted for the referenced table, the primary key column, and the columns that are necessary in the referenced table for viewing according to the defined display pattern.
- Columns that need to be in a defined display pattern in the table are given implicit viewing permissions.
- Permissions for database views of the **Proxy** type also apply to the underlying tables.
- For database views of the **ReadOnly** type, only the viewing permissions apply irrespective of other permissions.
- If a table or column is disabled due to preprocessor conditions, permissions are not determined for those tables and columns. The table or column is considered not to exist.

- If a permissions group is disabled due to preprocessor conditions, permissions are not taken into account for this permissions group. The permissions group is considered not to exist.

Example: Grouping permissions using permissions groups

The following example shows how to group permissions if the user is directly assigned in permissions groups and the permissions groups are not connected hierarchically.

A system user obtains permissions to the ADSAccount table through different permissions groups.

Permissions group	Viewable	Editable	Insertable	Deletable
A	1	1	1	1
B	0	0	0	0

In addition, it is granted permissions to the LDAPAccount table through these permissions groups.

Permissions group	Viewable	Editable	Insertable	Deletable
A	1	0	0	0
B	1	1	1	0

Therefore, the system user has effectively the following permissions:

Table	Viewable	Editable	Insertable	Deletable
ADSAccount	1	1	1	1
LDAPAccount	1	1	1	0

Example: Limiting conditions

A system user obtains viewing permissions to the Person table through different permissions groups.

Permissions group	Viewing condition	Viewing on columns
A		Lastname
B	Lastname like 'B%'	Lastname, Firstname, Entrydate
C	Lastname like 'Be%'	Lastname, Firstname, Gender
D	Lastname like 'D%'	Lastname

This results in the following permissions for the individual identities.

Person.Lastname	Visible Columns
Name1	Lastname
Name2	Lastname, Firstname, Entrydate
Name3	Lastname, Firstname, Gender
Name4	Lastname

Editing permissions groups

One Identity Manager provides permissions groups with a predefined user interface and special permissions for One Identity Manager schema's tables and columns. In certain isolated cases, it may be necessary to define custom permissions groups. You need custom permissions groups, for example, if:

- The default permissions groups grant too many permissions
- Selected default permissions groups are to be grouped to form a new permissions group
- Additional role-based permissions groups are required for the custom application roles
- Permissions for custom adjustments such as schema extensions, forms, or menu structures.

When the One Identity Manager database is installed using the Configuration Wizard, custom permissions groups that you can use are already created.

- For non role-based login, the **CCCViewPermissions** and **CCCEditPermissions** permission groups are created. Administrative system users are automatically added to these permissions groups.
- For role-based login, the **CCCViewRole** and **CCCEditRole** permission groups are created.

In the Designer, permissions groups are managed in the **Permissions > Permissions groups** category. Here you will find an overview of edit permissions and user interface components that are assigned to individual permissions groups. In addition, the system users are displayed to which the permissions groups are assigned.

Use the Designer to create and edit permissions groups with the User & Permissions Group Editor. The User & Permissions Group Editor displays the permissions groups in their hierarchy. Each permissions group is represented by a permissions group element. Each permissions group element has a tooltip. The contents of the tooltip is made up of the name and description of the permissions group.

You can run the following tasks:

- Edit permissions group main data
- Define new dependencies between permissions groups
- Copy permissions groups
- Create new permissions groups

Related topics

- [Predefined permissions groups and system users](#) on page 40
- [Permissions group dependencies](#) on page 47
- [Copying permissions groups](#) on page 48
- [Creating permissions groups](#) on page 49
- [Permissions group properties](#) on page 50
- [Managing permissions to program functions](#) on page 66

Dependencies between permissions groups

By structuring permissions groups hierarchically, permissions and user interface components can be passed down from one permissions group to another permissions group. This means that inheritance is top down within the hierarchy.

The following applies to permissions group dependencies:

- A role-based permissions group can inherit from role-based permissions groups and non role-based permissions groups.
- A non role-based permissions group can inherit from non role-based permissions groups. A non role-based permissions group must not inherit from role-based permissions groups.

Example:

Two permission groups are defined with the following permissions and user interface components.

Permissions group	Permissions	User interface
A	Viewable	Menu structures and forms
B	Editable	Task definitions

Permissions group B is assigned below permissions group A in the hierarchy and inherits from permissions group A. Consequently, a user of permissions group B has access to the viewing permissions and editing permissions as well as the menu structure, forms, and task definitions.

Related topics

- [Permissions group dependencies](#) on page 47

Permissions group dependencies

You edit dependencies between permissions groups in the hierarchical view of the User & Permissions Group Editor. Permissions groups that are higher up in the hierarchy are displayed further to the right in the User & Permissions Group Editor's hierarchical. When a permissions group is selected in the hierarchical view, dependencies to other permissions groups are marked in color thus showing the direction of inheritance.

Figure 1: Visual of the permissions group hierarchy (inheritance from right to left)



Table 21: Meaning of colors in the hierarchical representation

Color	Meaning
Blue	The selected permissions group.
Purple	This permissions group is a child of the selected permissions group and directly inherits from the selected permissions group.
Light purple	This permissions group inherits indirectly from the selected permissions group over the hierarchy.

Color	Meaning
Red	This permissions group is a child of the selected permissions group and directly inherits from the selected permissions group.
Light red	This permissions group passes inheritance indirectly to the selected permissions group over the hierarchy.
Green	This permissions group does not inherit or pass inheritance to the selected permissions group.

To specify dependencies of a permissions group

1. In the Designer, select the **Permissions > Permissions groups** category.
2. Select the permissions group and start the User & Permissions Group Editor using the **Edit permissions group** task.
3. In the hierarchical view of the permissions groups, select the permissions group and run one of the following actions.
 - Select the **Inherit permissions from** context menu and select the permissions groups from which the selected permissions group is to inherit.
 - Select the **Permissions inherited by** context menu and select the permissions groups to be included in the selected permissions group. Child permissions groups inherit permissions from the selected permissions group.
4. Select the **Database > Commit to database** and click **Save**.

Copying permissions groups

The User & Permissions Group Editor provides a wizard for copying permissions and the user interface of an existing permissions group to a new permissions group.

To copy a permissions group

1. In the Designer, select the **Permissions > Permissions groups** category.
2. Select the permissions group you want to copy and start the User & Permissions Group Editor with the **Edit permissions group** task.
3. Select the **Permissions groups > Copy permissions group** menu item.
4. On the start page of the wizard for copying permissions groups, click **Next**.
5. On the **Select permissions group** page, enter the following information:
 - **Select permissions group to copy:** The permissions group is pre-selected.
 - **Copy name:** Name of the new permissions group. A name suggestion is already entered that you can modify. Ensure that the permissions group name begins with the prefix **CCC**.

6. On the **Select copy options** page, specify which permissions group relations are to be copied. You can select multiple options. The following copy options are available.

Table 22: Copy options for permissions groups

Option	Description
Permissions	Enable this option to copy the table permissions and column permissions of the selected permissions group to the new permissions group.
User interface	Enable this option to copy the menu items, the forms and the task definitions of the selected permissions group to the new permissions group.
System user	Select this option if the system user should be copied to the new permissions group. NOTE: Predefined system users are not included in the new permissions group.

7. To start compiling, click **Next**.
The copying process may take some time.
8. The **Copy permissions group** page shows the individual copy steps and any error messages. If the copy action is complete, click **Next**.
9. To end the wizard, click **Finish** on the last page.

Related topics

- [Creating permissions groups](#) on page 49
- [Permissions group properties](#) on page 50
- [Permissions group dependencies](#) on page 47
- [Adding system users to permission groups](#) on page 54
- [Permissions for tables and columns](#) on page 56

Creating permissions groups

To create a permissions group

1. In the Designer, select the **Permissions** category.
2. Start the User & Permissions Group Editor with the **Show / edit permissions group** task.
3. Add a new permissions group using the **Permissions groups > New** menu item.
4. Edit the main data of the permissions group.
5. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Copying permissions groups](#) on page 48
- [Permissions group properties](#) on page 50
- [Permissions group dependencies](#) on page 47
- [Adding system users to permission groups](#) on page 54

Permissions group properties

Table 23: Properties of a permissions group

Property	Description
Permissions group	Name of the permissions group. Label custom permission groups with the prefix CCC .
Description	Detailed description of the permissions group's purpose.
Remarks	Text field for additional explanation.
Preprocessor condition	You can add a preprocessor condition to permissions groups. This means that the permissions group is only effective when the condition is met.
Permissions group binary pattern	The permissions group binary pattern is used to calculate effective system user permissions. It is provided by the DBQueue Processor.
Only use for role-based authentication	This group includes permissions, form assignments, menu items and program functions for role-based authentication. The permissions group can be assigned to One Identity Manager application roles and is assigned to dynamically determined system users. A direct assignment to non-dynamic system user is not permitted. NOTE: This function is available if the Identity Management Base Module is installed.

Related topics

- [Copying permissions groups](#) on page 48
- [Creating permissions groups](#) on page 49

Editing system users

One Identity Manager provides various system users whose permissions are matched to the various tasks. Create your own system users if required. Add the system users to

permissions groups, thereby granting the system users permissions for the tables and columns of the One Identity Manager schema and make the user interface available.

The system user's effective permissions that are found are not saved in the One Identity Manager schema, but are determined when logging into One Identity Manager tools and then they are loaded.

When installing the One Identity Manager database using the Configuration Wizard, create an administrative system user that is added to non role-based permissions groups and receives all the permissions of the **viadmin** default system user.

In the Designer, system users are displayed in the **Permissions > System users** category. This shows you an overview of the permissions groups that are assigned to each individual system user. Use the Designer to create and edit your system user in the User & Permissions Group Editor.

You can run the following tasks:

- Create new system users, such as an administrative system users or system users for service accounts
- Configure password settings for system users
- Add system users to permission groups
- Determine which identities use a system user

Related topics

- [Predefined permissions groups and system users](#) on page 40
- [Creating system users](#) on page 51
- [System users' passwords](#) on page 52
- [System user properties](#) on page 53
- [Adding system users to permission groups](#) on page 54
- [Which identities use the system user?](#) on page 55
- [Dynamic system user](#) on page 55

Creating system users

NOTE: You can create an administrative system user in User & Permissions Group Editor from the **Users > Create administrator** menu. Administrative system users are automatically added to all non role-based permissions groups.

To create a new system user

1. In the Designer, select the **Permissions** category.
2. Start the User & Permissions Group Editor with the **Show / edit permissions group** task.
3. Add a new system user using the **User > New** menu item.

4. Edit the system user's main data.
5. Add the system user to permissions groups.
6. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Predefined permissions groups and system users](#) on page 40
- [System users' passwords](#) on page 52
- [System user properties](#) on page 53
- [Adding system users to permission groups](#) on page 54
- [Dynamic system user](#) on page 55
- [Which identities use the system user?](#) on page 55
- [One Identity Manager authentication modules](#) on page 72

System users' passwords

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

If necessary, adjust the password policy to your requirements in the Designer. For more information about editing password policies, see the *One Identity Manager Operational Guide*.

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 identities, user accounts, or system users.

To prevent passwords expiring for service accounts, for example, in the Designer, you can enable the **Password never expires** (`DialogUser.PasswordNeverExpires`) option for the respective system users.

Related topics

- [System user properties](#) on page 53

System user properties

Table 24: Properties of a system user

Property	Description
System users	Name of the system user for logging in to the administration tools.
Password and password confirmation	Password with which the system logs into the administration tools.
Password last changed	Date of last password change.
Password never expires	Specifies whether the password never expires. Enable the option for service accounts, for example, to prevent the password from expiring. This option overwrites the maximum age of the password.
Remarks	Text field for additional explanation.
Read-only	Set the option if a system user is a member in several permissions groups, but has read-only permissions for the objects. This overrides all edit permissions that the system user is granted through memberships in permission groups.
Logins	Logins with which the system user can log in to the One Identity Manager tools. Enter the login in the form: Domain\User. This information is required if the Account based system user authentication module is used to log into the One Identity Manager tools.
Administrative user	<p>Specifies whether this is an administrative system user. Administrative system users are automatically added to all non role-based permissions groups.</p> <p>NOTE: You can create an administrative system user in the Designer with the User & Permissions Group Editor using the Create administrator menu.</p>
Service account	Specifies whether this is a system user that is used by a service account. This system user is not assigned to a permissions group but has all the permissions, tasks, and program functionality.
External password management	Specifies whether the system user password is determined by an external password management system. You cannot change the password in One Identity Manager. The determination of the system user password must be customized.
Disabled for direct login	Specifies whether the system user can be used for direct login. For example, enable the option for system users that are used for dynamic authentication modules to prevent direct logging in to One Identity Manager tools.

Related topics

- [Creating system users](#) on page 51
- [System users' passwords](#) on page 52
- [Configuration data for system user dynamic authentication](#) on page 114

Adding system users to permission groups

Add the system users to permissions groups, thereby granting permissions for the tables and columns of the One Identity Manager schema and make the user interface available.

NOTE:

- You cannot add system users to role-based permissions groups. Dynamic system users are calculated for role-based login.
- Administrative system users are automatically added to all non role-based permissions groups.
- The **QBM_BaseRights** permissions group defines the base rights that are required for a system user to log in to the One Identity Manager tools. This permissions group is always assigned implicitly.
- The **viadmin** system user has all of the specified permissions and the complete user interface. The system user implicitly receives the authorizations and user interface parts of the custom permissions groups.

A system user's memberships in permissions groups are displayed in the Designer in the User & Permissions Group Editor. Use the **Options > Display permissions group inheritance** menu to specify whether to display the direct and inherited memberships of permissions groups for a system user.

Figure 2: Memberships of a system user in permissions groups

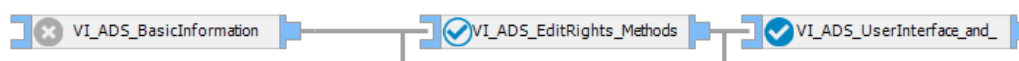


Table 25: Meaning of icons in the hierarchical display

Icon	Meaning
	The selected system user is not assigned to this permissions group.
	The selected system user is assigned to this permissions group.
	The selected system user is indirectly assigned to this permissions group.
	The selected system user is directly and indirectly assigned to this permissions group.

To assign a system user to a permissions group

1. In the Designer, select the **Permissions > System user** category.
2. Select a system user and start the User & Permissions Group Editor with the **Edit system user** task.
3. In the hierarchical view, select the permission group. By clicking on the icon, you add or delete the selected system user to or from a permissions group.
4. Select the **Database > Commit to database** and click **Save**.

TIP: To assign a system user to several permissions groups, use the **User > Permissions groups** menu.

Related topics

- [Dynamic system user](#) on page 55

Which identities use the system user?

Identities obtain a system user direct through their main data or dynamically through their One Identity Manager applications roles.

To display which identities are assigned to a system user

1. In the Designer, select the **Permissions > System user** category.
2. Select a system user and start the User & Permissions Group Editor with the **Edit system user** task.
3. Select the **View > One Identity Manager identities** menu item.

NOTE: You cannot change the assignments in this view.

Dynamic system user

Dynamic system users are used for logging into One Identity Manager tools with role-based authentication modules. First, the identity memberships in the One Identity Manager application roles are determined during login. Assignments of permissions groups to One Identity Manager application roles are used to determine which permissions groups apply to the identity. A dynamic system user is determined from these permissions groups that will be used for the identity's login.

NOTE: You cannot edit dynamic system users. If no role-based logins of identities that use dynamic system users are performed for some time, you should delete the dynamic system users for performance reasons. A new dynamic system user is created during the next role-based identity login.

To delete system users

- In the Designer, enable the **Common | DynamicUserLifetime** configuration parameter and enter the maximum retention period in days for dynamic system users.

If the configuration parameter is set, dynamic system users, whose retention period has expired, are deleted from the database as part of the daily maintenance tasks.

Permissions for tables and columns

In the Designer, you can edit permissions using the Permissions Editor. You can also simulate the permissions for the individual system users in the Permissions Editor.

With the Permissions Editor, you can:

- Grant permissions for custom tables and custom columns to custom permissions groups
- Grant permissions for predefined tables and predefined columns in the One Identity Manager schema to custom permissions groups
- Grant permissions for custom tables and custom columns to predefined permissions groups

Permissions of predefined permissions groups for predefined tables and predefined columns of the One Identity Manager schema cannot be changed

For custom schema extensions, use the Schema Extension program to specify permissions groups. A permissions group is given read and write permissions as well as a permissions group with read-only permissions. This make initial access to the custom schema extensions possible with the One Identity Manager administration tools.

Detailed information about this topic

- [Rules for determining the valid permissions for tables and columns](#) on page 43
- [Displaying permissions of a permissions group](#) on page 57
- [Displaying permissions for tables](#) on page 57
- [Editing table permissions](#) on page 58
- [Editing column permissions](#) on page 59
- [Copying table permissions and column permissions](#) on page 60
- [Simulating permissions for system users](#) on page 62

Displaying permissions of a permissions group

To display all permissions for a permission group

1. In the Designer, select the **Permissions** category.
2. Start the Permissions Editor using the **Edit permissions** task.
3. In the Permissions Editor toolbar in the **Permissions group** menu, select the permissions group whose permissions you want to display.

The tables and columns of the One Identity Manager schema and the permissions of the selected permissions group are displayed in the upper area of Permissions Editor. Use the following Permissions Editor options to adjust the layout.

- To display tables with permissions first, enable the **Options > Permissions** sort order menu.
- To display disabled tables and columns, enable the **Options > Show disabled tables** menu.
- To use the display names of the tables and columns, enable the **Options > Display name** menu.
- To limit the display of the tables, use the **Show system tables**, **Show non-system tables**, and **Show all tables** menu items in the **Options** menu. Alternatively, use the **Define filter** or **Manage filters** menu items to define your own user-defined filters for displaying the tables and columns.

For more information about working with user-defined filters in the Designer, see *One Identity Manager User Guide for One Identity Manager Tools User Interface*.

Displaying permissions for tables

In the Permissions Editor, the **Summary of all permissions** view displays the permissions groups that have permissions for the selected column. The permissions in this view cannot be edited.

NOTE: To display the **Summary of all permissions** view, go to the Permissions Editor and enable **View > Object permissions** menu. The view is displayed in the lower area of the Permissions Editor.

To display all permissions for a table and its columns

1. In the Designer, select the table in the **Permissions > By tables** category.
2. Start the Permissions Editor using the **Edit permissions for table** task.

The **Summary of all permissions** view displays the permissions groups that have permissions for the selected table.

| **TIP:** To display a permissions filter completely, click a condition in the view.

3. (Optional) To display all the column permissions, open the table entry in the upper part of the Permissions Editor and select a column.

The **Summary of all permissions** view displays the permissions groups that have permissions for the selected column.

Editing table permissions

Use the table permissions to grant permissions to display, insert, edit, and delete the objects. You can define conditions to further limit the permissions for the objects. You can use the conditions, for example, to link the editability of the identities to their last names. For instance, users can be given read-only access to the identities whose last names begin with A-F, whereas they can edit identities with last names beginning with G-Z.

| **NOTE:** Permissions are always edited in the Permissions Editor for the permissions group that you selected in the Permissions Editor toolbar in the **Permissions group** menu. If you wish to grant permissions for another permissions group, first select this permissions group in the menu and then edit the permissions.

To edit the table permissions for a permissions group

1. In the Designer, select the **Permissions** category.
 2. Start the Permissions Editor using the **Edit translation in database** task.
 3. In the Permissions Editor toolbar in the **Permissions group** menu, select the permissions group for which you want to grant the permissions.
 4. Select the table at the top of the Permissions Editor.
- | **TIP:** Use **Shift + select** or **Ctrl + select** to select multiple tables.
5. In the **Permissions** section, edit the permissions for the permissions group.

- To insert new permissions, select the **New** context menu and enable the associated check boxes. Grant the following permissions:
 - **Viewable:** The table data is displayed.
 - **Insertable:** New data can be added to the table.
 - **Editable:** Table data can be edited.
 - **Deletable:** Table data can be deleted

| **NOTE:** If you grant the **Insertable**, **Editable**, or **Deletable** permissions, the **Viewable** permission is also granted.

- To withdraw permissions, disable the associated checkbox.
 - Use the **Delete** context menu, to withdraw all permissions from a table.
6. (Optional) To specify other conditions for table permissions, go to the lower part of the Permissions Editor and switch to the **Group permissions for table** view and select the **Permissions filter** tab.

NOTE: You can only define permissions filters for the tables that map application data.

- Enter the conditions as valid WHERE clauses for database queries. You can enter the following permissions filters.
 - **Viewing Condition:** Limiting condition for displaying data sets.
 - **Edit condition:** Limiting condition for editing data sets.
 - **Insert condition:** Limiting condition for inserting data sets.
 - **Deletion condition:** Limiting condition for deleting data sets.

Example: Permissions filter

A user should be able to see all identities, but only edit the identities whose last names begin with B. Specify the limiting edit condition as follows, for example:

```
Lastname like 'B%'
```

TIP: Use the **SQL check** button to test the condition. This checks the syntax. The number of objects that match the condition is returned.

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

Related topics

- [Editing column permissions](#) on page 59
- [Copying table permissions and column permissions](#) on page 60

Editing column permissions

IMPORTANT:

- If you grant permissions to columns, you must also grant the permissions to the tables. For example, a column is only viewable if the table is also viewable.
- To insert objects into a table, the **Insert** permission is required at least for the mandatory fields in the table.
- If you grant **Insert** or **Edit** permissions, **View** permissions are also granted.
- Column definition allows you to use scripts to conditionally display or edit a column. For example, in this way you can control whether or not a column, on a main data form in the Manager, is displayed or can be edited only if another column has a specific value. The script does not change the user's permissions but simply the behavior if the object is loaded in one of the One Identity Manager tools. For more

information about editing column definitions, see the *One Identity Manager Configuration Guide*.

NOTE: Permissions are always edited in the Permissions Editor for the permissions group that you selected in the Permissions Editor toolbar in the **Permissions group** menu. If you wish to grant permissions for another permissions group, first select this permissions group in the menu and then edit the permissions.

To edit column permissions for a permissions group

1. In the Designer, select the **Permissions** category.
2. Start the Permissions Editor using the **Edit permissions** task.
3. In the Permissions Editor toolbar in the **Permissions group** menu, select the permissions group for which you want to grant the permissions.
4. Select the table at the top of the Permissions Editor and select the column.
TIP: Use **Shift + select** or **Ctrl + select** to select multiple columns.
5. In the **Permissions** section, edit the permissions for the permissions group.
 - To insert new permissions, select the **New** context menu and enable the associated check boxes. Grant the following permissions:
 - **Viewable:** The column is displayed.
 - **Editable:** The value in the column can be changed.
 - **Insertable:** The value of the column can be edited when adding a new data record. Once the data record has been saved it can no longer be edited.
 - To withdraw permissions, disable the associated checkbox.
 - Use the **Delete** context menu item, to delete all permissions from a column.
6. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Editing table permissions](#) on page 58
- [Copying table permissions and column permissions](#) on page 60

Copying table permissions and column permissions

To transfer the permissions of a permissions group quickly from one table to another table, you can copy the table permissions and column permissions. Two methods are provided in the Permissions Editor to do this:

- **Copy** and **Insert**: This method copies permissions of the source table (source column) of a permissions group. The permissions are copied for the permissions group that you selected in the Permissions Editor toolbar in the **Permissions group** menu.

All copied permissions are inserted for the target table (target column). Already existing permissions for the target table (target column) remain the same.

- **Copy all permissions** and **Paste all permissions**: This method copies all source table (source column) permissions. The initial selection of the permissions group in the Permissions Editor makes no difference here. All permissions from all permissions groups for the source table (source column) are applied.

All copied permissions are inserted for the target table (target column). Existing permissions for target table (target column) that do not exist for the source table (source column) are removed from the target table (target column).

To copy permissions of a permission group

1. In the Designer, select the **Permissions** category.
2. Start the Permissions Editor using the **Edit permissions** task.
3. In the Permissions Editor toolbar in the **Permissions group** menu, select the permissions group for which you want to grant the permissions.
4. To transfer the table permissions.
 - a. Select the table at the top of the Permissions Editor from which you want to transfer the permissions.
 - b. Use the **Copy** context menu item to copy the permissions to the clipboard.
 - c. Select the table at the top of the Permissions Editor from which you want to transfer the permissions.
 - d. Use the **Insert** context menu to insert the permissions.
 - e. If necessary, repeat step c) and d) for other tables.
5. To transfer the column permissions
 - a. Select the table at the top of the Permissions Editor and select the column from which you want to transfer permissions.
 - b. Use the **Copy** context menu to copy the permissions.
 - c. Select the table at the top of the Permissions Editor and select the column for which you want to transfer permissions.
 - d. Use the **Insert** context menu to insert the permissions.
 - e. If necessary, repeat step c) and d) for other columns.
6. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Editing table permissions](#) on page 58
- [Editing column permissions](#) on page 59

Simulating permissions for system users

By simulating the permissions in the Permissions Editor, you can see which permissions a system user has based on their permissions group. You can specify which permissions groups of a system user to include in the simulation. The result displayed shows which of the selected permissions groups has which table permissions and column permissions. Effective permissions for the system user are also displayed.

NOTE: Simulation mode remains active until you end it. In simulation mode, you can edit permissions group permissions and update simulation data.

To run a simulation:

1. In the Designer, select the **Permissions** category.
2. Start the Permissions Editor using the **Edit translation in database** task.
3. From the **Simulation > Start simulation** menu, start the simulation wizard.
4. On the start page of the wizard, click **Next**.
5. On the **Simulation base configuration** page, select the following data.
 - **User:** Select the system user whose permissions you want to simulate.
 - **Direct groups:** Use this button to select all permissions groups that are directly assigned to the system user.
 - **All groups:** Use this button to select all permissions groups that are directly assigned to the system user as well as all permissions groups that the system user inherits indirectly.
 - **Permissions groups:** Select individual permissions groups directly. Use **Ctrl + select** to select multiple permissions groups.
6. On the **Simulation configuration** page, specify the tables for which the permissions are simulated.
 - In the **Selected tables** pane, all tables of the One Identity Manager schema are selected. If necessary, limit the selection to individual tables. Click **None** to undo the selection. Use **Shift + select** to select individual tables.
 - Using the **Context table** menu, you can specify a table from which you can view the resulting implicit permissions for the foreign key columns display values.

Example:

Viewing permissions are granted on the UID_Org column of the Person table. As a result, viewing permissions are implicitly assigned to columns of the Org table that are used as a display template, for example, Org.Ident_Org.

To simulate this example, select the **Person** table under Context table and the **Org** table under Selected tables.

7. The processing progress of the simulation is displayed on the **Simulation** page. The simulation process can take some time.
8. To end the wizard, click **Finish** on the last page.

After you complete simulation wizard, the system user's effective table permissions and column permissions are displayed in the upper part of the Permissions Editor in the **Simulation** view.
9. To determine which table permission or column permission results from which of the system user's permissions groups, select the table or column in the upper part of the Permissions Editor.

The permissions and permissions groups are displayed in the **Permissions simulation** view in the lower part of the Permissions Editor.
10. To end the simulation mode, select the **Simulation > End simulation** menu.

The simulation data is deleted and the **Permissions simulation** view is closed.

Displaying permissions for objects

| NOTE: The Manager must be running in expert mode to show object properties. You can display object properties and permissions in One Identity Manager tools.

To view an object's permissions

1. Select the object and open the **Properties** context menu.
2. Select the **Permissions** tab.

On the **Permissions** tab, based on the permissions groups, you see what permissions apply to an object. The first entry shows the basic permissions for the table. The permissions for this particular object are displayed beneath that. The other entries show the column permissions.

| TIP: Double-click the table entry, the object entry, or a column entry to display the permissions group from which the permissions were determined.

Table 26: Icon used for permissions

Icon	Meaning
✓	Permissions exist.
•	Permissions have been removed by the object layer.
☑	Permissions limited by conditions.

Displaying permissions for the current user

To get more information about the current user

- To display user information, double-click the  icon in the program status bar.

Table 27: Extra information about the current user

Property	Meaning
System users	Name of system user
Authenticated by	Name of the authentication module used for logging in.
Identity UID (UserID)	Unique ID for the current user's identity if an identity-based authentication module is used to log in.
SQL access level	Access level of the database server used to log in.
Read-only	The system user has only has read permissions. Modification to data are not possible.
Dynamic user	The current user uses a dynamic system user. Dynamic system users are applied when a role-based authentication module is used.
Administrative user	The current user uses an administrative system user.
Remarks	More details about the system user in use.
Permissions group	Permissions groups that are assigned to the system user. The permissions groups determine the user's user interface and object permissions.
Program functions	Program functions assigned to the system user The menu items and functions available depend on the program functions.

Assigning role-based permissions groups to an applications

If you assign a permissions group to an application, the permissions of the group apply only to this application. When a user logs on to the application, they receive the permissions of the permissions group in addition to their own permissions.

To assign a role-based permissions group to an application

1. In the Designer, select the **Permissions > Permissions groups > Role based permissions groups** category.
2. Select **View > Select table relations** and enable the `DialogGroupInProductLimited` table.
3. In the List Editor, select the permissions group.
4. Assign the application in the **Applications** edit view.
5. Select the **Database > Commit to database** and click **Save**.

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

Managing permissions to program functions

Program functions are part of the permission model in One Identity Manager. They allow you to enable and disable functionality. Program functions are not assigned to single users but to permissions groups. The set of program functions defined for a user is determined by their permissions groups and the program functions contained in them.


One Identity Manager tools can only be started if the user has the relevant program function permissions. Furthermore, some functions in the One Identity Manager tools are available only if the program functions are assigned to the current user. This includes data export from the Manager, calling the SQL Editor in the Designer or showing DBQueue Processor information in all programs, as examples.

Detailed information about this topic

- [Displaying permissions for the current user on page 64](#)
- [Assigning program functions to permissions groups on page 67](#)
- [Permissions for running scripts on page 67](#)
- [Permissions for running methods on page 68](#)
- [Permissions for triggering processes on page 69](#)
- [Modifying permissions for running actions in the Launchpad on page 70](#)
- [Program functions for starting the One Identity Manager tools on page 150](#)

Displaying the current user's program functions

To identify the program functions available to the current user:

- To display user information, double-click the icon in the program status bar 
The **Program functions** tab shows the program functions that are available.

Assigning program functions to permissions groups

To assign a program function to permissions groups

1. In the Designer, select the **Permissions > Program functions** category.
2. Select the **View > Select table relations** menu item and enable the DialogGroupHasFeature table.
3. In the List Editor, select the program function.
4. Assign the permissions group in the **Permissions groups** edit view.
5. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Adding system users to permission groups](#) on page 54

Permissions for running scripts

The basic permissions for running scripts are granted to the logged in user by the **Common_StartScripts** program function.

If a script is assigned a program function (QBMScriptHasFeature table), users can only run this script if they have the necessary permissions groups. An error occurs if the user does not own this program function and tries to run it.

To control how a script is run using a program function

1. Create a new program function.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **Object > New** menu item.
 - c. Enter the following information:
 - **Program function:** Name of the program function.
 - **Description:** Short description of the program function.
 - **Function group:** Property for grouping program functions.
2. Connect the program function with the scripts that the user are allowed to trigger.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **View > Select table relations** menu item and enable the QBMScriptHasFeature table.

- c. In the List Editor, select the newly created program function.
 - d. In the **Scripts** edit view, assign the scripts.
3. Assign the required program functions to the custom permissions group whose systems users will run these scripts.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **View > Select table relations** menu item and enable the DialogGroupHasFeature table.
 - c. In the List Editor, select your newly created program function.
 - d. In the List Editor, use **Ctrl + select** to select your new program function and the **Common_StartScripts** program function.
 - e. Assign the permissions group in the **Permissions groups** edit view.
4. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Editing permissions groups](#) on page 45

Permissions for running methods

If a task definition is assigned a program function (QBMMethodHasFeature table) users can only run this task if they have the necessary permissions groups. An error occurs if the user does not own this program function and tries to run it.

To make a task definition available to users using a program function

1. Create a new program function.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **Object > New** menu item.
 - c. Enter the following information:
 - **Program function:** Name of the program function.
 - **Description:** Short description of the program function.
 - **Function group:** Property for grouping program functions.
2. Connect the program function with the task definition events that the user will trigger.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **View > Select table relations** menu item and enable the QBMMethodHasFeature table.

- c. In the List Editor, select the newly created program function.
 - d. In the **Tasks** edit view, assign the task definitions.
3. Assign the program functions to the custom permissions group whose systems users will run these scripts.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **View > Select table relations** menu item and enable the DialogGroupHasFeature table.
 - c. In the List Editor, select your newly created program function.
 - d. Assign the permissions group in the **Permissions groups** edit view.
4. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Editing permissions groups](#) on page 45

Permissions for triggering processes

The basic permissions for triggering processes are granted to the logged in user by the **Common_TriggerEvents** program function.

In One Identity Manager, triggering of events on stored processes is linked to the permissions concept. Users can only trigger events on objects like this if they own edit permissions for them. This can lead to table users who only have viewing permissions not being able to trigger additional events for processes.

In this case, it is possible to connect the object events (QBMEvent table) with a program function (QBMFeature table). An event (JobEventGen table), which is defined for a process, is linked with an object event (JobEventGen.UID_QBMEvent column). The object events are linked to a program function (QBMEventHasFeature table). Users with this program function can trigger the object event and therefore the process too independent of their permissions.

TIP: The **Common_TriggerSpecificEvents** program function allows you to trigger specific events from the front-end. You can assign this program function to custom object events that any user can trigger. The program function is allocated to the **QBM_BaseRigt** permissions group.

To control triggering a process through a program function

1. Create a new program function.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **Object > New** menu item.

- c. Enter the following information:
 - **Program function:** Name of the program function.
 - **Description:** Short description of the program function.
 - **Function group:** Property for grouping program functions.
2. Connect the program function with object events that the user will trigger.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **View > Select table relations** menu item and enable the QBMEventHasFeature table.
 - c. In the List Editor, select the newly created program function.
 - d. In the **Object events** edit view, assign the object events.
3. Assign the required program functions to the custom permissions group whose systems users will trigger these events.
 - a. In the Designer, select the **Permissions > Program functions** category.
 - b. Select the **View > Select table relations** menu item and enable the DialogGroupHasFeature table.
 - c. In the List Editor, use **Ctrl + select** to select your new program function and the **Common_TriggerEvents** program function.
 - d. Assign the permissions group in the **Permissions groups** edit view.
4. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Editing permissions groups](#) on page 45

Modifying permissions for running actions in the Launchpad

One Identity Manager supplies a number of Launchpad actions that you can use to start applications by using the Launchpad. You can also start your own applications over the Launchpad.

If some actions in the Launchpad should not be made available to all users, you can manage the permissions by assigning Launchpad actions to program functions (QBMLaunchActionHasFeature table). Only tasks containing actions that the user's program function permissions permit him to run are shown in the Launchpad.

To assign a program function to Launchpad actions

1. In the Designer, select the **Permissions > Program functions** category.
2. Select the **View > Select table relations** menu item and enable the QBMLaunchActionHasFeature table.
3. In the List Editor, select the program function.
4. In the **Launchpad action** edit view, assign the actions.
5. Select the **Database > Commit to database** and click **Save**.

One Identity Manager authentication modules

One Identity Manager uses different authentication modules for logging in to administration tools. Authentication modules identify the system users to be used and load the user interface and database resource editing permissions depending on their permission group memberships.

- The permissions assigned to the system user are found from the permissions groups for logging into One Identity Manager tools with an authentication module that expects a defined system user.
- Dynamic system users are used for logging into One Identity Manager tools with role-based authentication modules. First, the identity memberships in the One Identity Manager application roles are determined during login. Assignments of permissions groups to One Identity Manager application roles are used to determine which permissions groups apply to the identity. A dynamic system user is determined from these permissions groups that will be used for the identity's login.

Before you can use an authentication module for logging on, the following prerequisites must be fulfilled:

1. The authentication module must be enabled.
2. The authentication module must be assigned to the application.
3. The assignment of the authentication module to the application must be enabled.

This allows you to log in to the assigned application using this authentication module. Ensure that users found through the authentication module also have the required program function to use the program.

NOTE: After the initial schema installation, only the **System user** and **Component authenticator** authentication modules and the role-based authentication modules are enabled in One Identity Manager.

Use non role-based authentication modules to log in to the Designer. Role-based authentication modules for logging in to the Designer are not supported.

NOTE: Authentication modules are defined in the One Identity Manager modules and are not available until the modules are installed.

Related topics

- [Enabling authentication modules](#) on page 106
- [Assigning authentication modules to applications](#) on page 107
- [Disabling or enabling authentication modules for applications](#) on page 108

System users

NOTE: This authentication module is available if the Configuration Module is installed.

Credentials	The system user's identifier and password.
Prerequisites	<ul style="list-style-type: none">• The system user with permissions exists in the One Identity Manager database.
Set as default	Yes
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	No
Remarks	<p>The user interface and the permissions are loaded through the system user.</p> <p>Data modifications are attributed to the system user.</p>

IMPORTANT: The **viadmin** system user is available by default. The system user has the predefined user interface and access permissions to database resources. You must not use or change the user interface and the permissions structure of the system user in live systems because this system user is overwritten with each schema update as it is from a system user template.

TIP: Create your own system user with the appropriate permissions. This can be done on initial installation of the One Identity Manager database. This system user can compile an initial One Identity Manager database and can be used to log into the administration tools for the first time.

Generic single sign-on (role-based)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

Credentials	The authentication module uses the login data of the user currently logged in on the workstation.
Prerequisites	<ul style="list-style-type: none"> • The identity exists in the One Identity Manager database. • The identity is assigned at least one application role. • The user account exists in the One Identity Manager database and the identity is entered in the user account's main data.
Set as default	No
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>One Identity Manager searches for the user account according to the configuration and finds the identity assigned to the user account.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none"> • If this configuration parameter is set, the identity's main identity is used for authentication. • If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p> <p>A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.</p> <p>Changes to the data are assigned to the logged in identity.</p>

Modify the following configuration parameters in the Designer to implement the authentication module.

Table 28: Configuration parameters for the authentication module

Configuration parameter	Meaning
QER Person GenericAuthenticator	Specifies whether authentication through single sign-on is supported.

Configuration parameter	Meaning
QER Person GenericAuthenticator SearchTable	Table in the One Identity Manager schema which stores the user information. The table must contain a foreign key with the name UIDPerson (or CCC_UID_Person) that references the Person table. Example: ADSAccount
QER Person GenericAuthenticator SearchColumn	Column from the One Identity Manager table (SearchTable) that is used to search for user name of the current user. Example: CN
QER Person GenericAuthenticator EnabledBy	Pipe () delimited list of Boolean columns from the One Identity Manager table (SearchTable) enabled by the user account for the login.
QER Person GenericAuthenticator DisabledBy	Pipe () delimited list of Boolean columns from the One Identity Manager table (SearchTable) disabled by the user account for the login. Example: AccountDisabled

Identity

NOTE: This authentication module is available if the Identity Management Base Module is installed.

Credentials	Identity's central user account and password.
Prerequisites	<ul style="list-style-type: none"> • The system user with permissions exists in the One Identity Manager database. • The identity exists in the One Identity Manager database. • The central user account is entered in the identity main data. • The system user is entered in the identity's main data. • The system user password is entered in the identity main data.
Set as default	Yes
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication

configuration parameter controls which identity is used for authentication.

- If this configuration parameter is set, the identity's main identity is used for authentication.
- If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

The user interface and permissions are loaded through the system user that is directly assigned to the logged in identity.

Changes to the data are assigned to the logged in identity.

Identity (role-based)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

Credentials	Identity's central user account and password.
Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• The central user account is entered in the identity main data.• The system user password is entered in the identity main data.• The identity is assigned at least one application role.
Set as default	Yes
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.• If this configuration parameter is set, the identity's subidentity is

used for authentication.

NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.

Changes to the data are assigned to the logged in identity.

Identity (dynamic)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

Credentials	Identity's central user account and password.
Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• The central user account is entered in the identity main data.• The system user password is entered in the identity main data.• The configuration data for dynamically determining the system user is defined in the application. Thus, an identity can, for example, be assigned a system user dynamically depending on their department membership.
Set as default	Yes
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.• If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be

able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

The application configuration data is used to find a system user, which is automatically assigned to the identity. The user interface and permissions are loaded through the system user that is dynamically assigned to the logged in identity.

Changes to the data are assigned to the logged in identity.

Related topics

- [Configuration data for system user dynamic authentication](#) on page 114

User account

NOTE: This authentication module is available if the Identity Management Base Module is installed.

Credentials	The authentication module uses the Active Directory login data of the user currently logged in on the workstation.
Prerequisites	<ul style="list-style-type: none">• The system user with permissions exists in the One Identity Manager database.• The identity exists in the One Identity Manager database.• Permitted logins are entered in the identity main data. The logins are expected in the form: domain\user.• The system user is entered in the identity's main data.
Set as default	No
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>All identity logins saved in the One Identity Manager database are found. The identity whose login data matches that of the current user is used for logging in.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p>

- If this configuration parameter is set, the identity's main identity is used for authentication.
- If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

The user interface and permissions are loaded through the system user that is directly assigned to the identity found.

Data modifications are attributed to the current user account.

User account (role-based)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

Credentials	The authentication module uses the Active Directory login data of the user currently logged in on the workstation.
Prerequisites	<ul style="list-style-type: none"> • The identity exists in the One Identity Manager database. • Permitted logins are entered in the identity main data. The logins are expected in the form: domain\user. • The identity is assigned at least one application role.
Set as default	No
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>All identity logins saved in the One Identity Manager database are found. The identity whose login data matches that of the current user is used for logging in.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none"> • If this configuration parameter is set, the identity's main identity is used for authentication.

- If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.

Data modifications are attributed to the current user account.

User account (manual input/role-based)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

Credentials	Login name and password for registering with Active Directory. You do not have to enter the domain.
Prerequisites	<ul style="list-style-type: none"> • The identity exists in the One Identity Manager database. • Permitted logins are entered in the identity main data. The logins are expected in the form: domain\user. • The identity is assigned at least one application role. • Domains permitted for login are entered in the TargetSystem ADS AuthenticationDomains configuration parameter. <p>NOTE: This configuration parameter is available if the Active Directory Module is installed.</p>
Set as default	Yes
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	All identity logins saved in the One Identity Manager database are found. The identity whose login data matches that of the current user is used for logging in. This takes into account the list of permitted Active Directory domains.

If an identity has a main identity or several subidentities, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which identity is used for authentication.

- If this configuration parameter is set, the identity's main identity is used for authentication.
- If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.

Data modifications are attributed to the current user account.

Account based system user

NOTE: This authentication module is available if the Configuration Module is installed.

Credentials	The authentication module uses the Active Directory login data of the user currently logged in on the workstation.
Prerequisites	<ul style="list-style-type: none">• The system user with permissions exists in the One Identity Manager database.• Permitted logins are entered in the system user's main data. The logins are expected in the form: domain\user.
Set as default	No
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	No
Remarks	<p>All system user logins saved in the One Identity Manager database are found. The system user whose login data matches that of the current user is used for logging in.</p> <p>The user interface and the permissions are loaded through the system user.</p> <p>Data modifications are attributed to the current user account.</p>

Active Directory user account

NOTE: This authentication module is available if the Active Directory Module is installed.

Credentials	The authentication module uses the Active Directory login data of the user currently logged in on the workstation.
Prerequisites	<ul style="list-style-type: none">• The system user with permissions exists in the One Identity Manager database.• The identity exists in the One Identity Manager database.• The system user is entered in the identity's main data.• The Active Directory user account exists in the One Identity Manager database and the identity is entered in the user account's main data.
Set as default	Yes
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>The appropriate user account is found in the One Identity Manager database through the user's SID and the domain given at login. One Identity Manager finds the identity assigned to the user account.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.• If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p> <p>The user interface and permissions are loaded through the system user that is directly assigned to the identity found. If a system user is not assigned to the identity, the system user from the SysConfig Logon DefaultUser configuration parameter is used.</p> <p>Data modifications are attributed to the current user account.</p>

NOTE: If the **Connect automatically** option is set, authentication is no longer necessary for subsequent logins.

Active Directory user account (role-based)

NOTE: This authentication module is available if the Active Directory Module is installed.

Credentials	The authentication module uses the Active Directory login data of the user currently logged in on the workstation.
Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• The identity is assigned at least one application role.• The Active Directory user account exists in the One Identity Manager database and the identity is entered in the user account's main data.
Set as default	Yes
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>The appropriate user account is found in the One Identity Manager database through the user's SID and the domain given at login. One Identity Manager finds the identity assigned to the user account.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.• If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p> <p>A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.</p> <p>Data modifications are attributed to the current user account.</p>

NOTE: If the **Connect automatically** option is set, authentication is no longer necessary for subsequent logins.

Active Directory user account (manual input)

NOTE: This authentication module is available if the Active Directory Module is installed.

Credentials	Login name and password for registering with Active Directory. You do not have to enter the domain.
Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• The Active Directory user account exists in the One Identity Manager database and the identity is entered in the user account's main data.• Domains permitted for login are entered in the TargetSystem ADS AuthenticationDomains configuration parameter.• The configuration data for dynamically determining the system user is defined in the application. Thus, an identity can, for example, be assigned a system user dynamically depending on their department membership.
Set as default	No
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>The user's identity is determined from a predefined list of permitted Active Directory domains. The corresponding user account and identity are determined in the One Identity Manager database, which the user account is assigned to.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.• If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be

able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

The application configuration data is used to find a system user, which is automatically assigned to the identity. The user interface and permissions are loaded through the system user that is dynamically assigned to the logged in identity.

Data modifications are attributed to the current user account.

Active Directory user account (manual input/role-based)

NOTE: This authentication module is available if the Active Directory Module is installed.

Credentials	Login name and password for registering with Active Directory. You do not have to enter the domain.
Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• The identity is assigned at least one application role.• The Active Directory user account exists in the One Identity Manager database and the identity is entered in the user account's main data.• Domains permitted for login are entered in the TargetSystem ADS AuthenticationDomains configuration parameter.
Set as default	Yes
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>The user's identity is determined from a predefined list of permitted Active Directory domains. The corresponding user account and identity are determined in the One Identity Manager database, which the user account is assigned to.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.

- If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.

Data modifications are attributed to the current user account.

Active Directory user account (dynamic)

NOTE: This authentication module is available if the Active Directory Module is installed.

Credentials	The authentication module uses the Active Directory login data of the user currently logged in on the workstation.
Prerequisites	<ul style="list-style-type: none"> • The identity exists in the One Identity Manager database. • The Active Directory user account exists in the One Identity Manager database and the identity is entered in the user account's main data. • The configuration data for dynamically determining the system user is defined in the application. Thus, an identity can, for example, be assigned a system user dynamically depending on their department membership.
Set as default	No
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>The appropriate user account is found in the One Identity Manager database through the user's SID and the domain given at login. One Identity Manager finds the identity assigned to the user account.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p>

- If this configuration parameter is set, the identity's main identity is used for authentication.
- If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

The application configuration data is used to find a system user, which is automatically assigned to the identity. The user interface and permissions are loaded through the system user that is dynamically assigned to the logged in identity.

Data modifications are attributed to the current user account.

NOTE: If the **Connect automatically** option is set, authentication is no longer necessary for subsequent logins.

Related topics

- [Configuration data for system user dynamic authentication](#) on page 114

LDAP user account (role-based)

NOTE: This authentication module is available if the LDAP Module is installed.

Credentials	<p>Login name, identifier, distinguished name or user ID of an LDAP user account.</p> <p>LDAP user account's password.</p>
Prerequisites	<ul style="list-style-type: none"> • The identity exists in the One Identity Manager database. • The identity is assigned at least one application role. • The LDAP user account exists in the One Identity Manager database and the identity is entered in the user account's main data. • The configuration data for dynamically determining the system user is defined in the application. Thus, an identity can, for example, be assigned a system user dynamically depending on their department membership.
Set as default	No
Single sign-on	No
Front-end login allowed	Yes

Web Portal login allowed	Yes
Remarks	<p>If you log in using a login name, identifier, or user ID, the corresponding user account is determined in the One Identity Manager database through the domain. The domains permitted for logging in are entered in the TargetSystem LDAP Authentication RootDN configuration parameter and the TargetSystem LDAP AuthenticationV2 RootDN configuration parameter. If log in uses a distinguished name, the LDAP user account is determined that uses this distinguished name. One Identity Manager finds the identity assigned to the LDAP user account.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none"> • If this configuration parameter is set, the identity's main identity is used for authentication. • If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p> <p>A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.</p> <p>Data modifications are attributed to the current user account.</p>

In the Designer, modify the following configuration parameters to implement the authentication module.

Table 29: Configuration parameters for the authentication module

Configuration parameter	Meaning
TargetSystem LDAP Authentication	Allows configuration of the LDAP authentication module.
TargetSystem LDAP Authentication Authentication	<p>Authentication mechanism. Permitted values are Secure, Encryption, SecureSocketsLayer, ReadonlyServer, Anonymous, FastBind, Signing, Sealing, Delegation, and ServerBind. The value can be combined with commas (,). For more information about authentication types, see the MSDN Library.</p> <p>Default: ServerBind</p>

Configuration parameter	Meaning
TargetSystem LDAP Authentication Port	Communications port on the server. Default: 389
TargetSystem LDAP Authentication RootDN	Pipe () delimited list of root domains to be used to find the user account for authentication. Syntax: DC=<MyDomain> DC=<MyOtherDomain> Example: DC=Root1,DC=com DC=Root2,DC=de
TargetSystem LDAP Authentication Server	Name of the LDAP server.
TargetSystem LDAP AuthenticationV2	Allows configuration of the LDAP authentication module.
TargetSystem LDAP AuthenticationV2 AcceptSelfSigned	Specifies whether self-signed certificates are accepted.
TargetSystem LDAP AuthenticationV2 Authentication	Authentication method for logging in to LDAP. The following are permitted: <ul style="list-style-type: none"> • Basic: Uses default authentication. • Negotiate: Uses Negotiate authentication from Microsoft. • Kerberos: Uses Kerberos authentication. • NTLM: Uses Windows NT Challenge/Response (NTLM) authentication. Default: Basic For more information about authentication types, see the MSDN Library .
TargetSystem LDAP AuthenticationV2 ClientTimeout	Client timeout in seconds.
TargetSystem LDAP AuthenticationV2 Port	Communications port on the server. Default: 389
TargetSystem LDAP AuthenticationV2 ProtocolVersion	Version of the LDAP protocol. The values 2 and 3 are permitted. Default: 3

Configuration parameter	Meaning
TargetSystem LDAP AuthenticationV2 RootDN	<p>Pipe () delimited list of root domains to be used to find the user account for authentication.</p> <p>Syntax:</p> <p>DC=<MyDomain> DC=<MyOtherDomain></p> <p>Example:</p> <p>DC=Root1,DC=com DC=Root2,DC=de</p>
TargetSystem LDAP AuthenticationV2 Security	Connection security. Permitted values are None , SSL and STARTTLS .
TargetSystem LDAP AuthenticationV2 Server	Name of the LDAP server.
TargetSystem LDAP AuthenticationV2 UseSealing	Specifies whether sealing is enabled.
TargetSystem LDAP AuthenticationV2 UseSigning	Specifies whether signing is enabled.
TargetSystem LDAP AuthenticationV2 VerifyServerCertificate	Specifies whether to check the server certificate when encrypting with SSL.

LDAP user account (dynamic)

NOTE: This authentication module is available if the LDAP Module is installed.

Credentials	<p>Login name, identifier, distinguished name or user ID of an LDAP user account.</p> <p>LDAP user account's password.</p>
Prerequisites	<ul style="list-style-type: none"> The identity exists in the One Identity Manager database. The LDAP user account exists in the One Identity Manager database and the identity is entered in the user account's main data. The configuration data for dynamically determining the system user is defined in the application. Thus, an identity can, for example, be assigned a system user dynamically depending on their department membership.

Set as default	No
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>If you log in using a login name, identifier, or user ID, the corresponding user account is determined in the One Identity Manager database through the domain. The domains permitted for logging in are entered in the TargetSystem LDAP Authentication RootDN configuration parameter and the TargetSystem LDAP AuthenticationV2 RootDN configuration parameter. If log in uses a distinguished name, the LDAP user account is determined that uses this distinguished name. One Identity Manager finds the identity assigned to the LDAP user account.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none"> • If this configuration parameter is set, the identity's main identity is used for authentication. • If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p> <p>The application configuration data is used to find a system user, which is automatically assigned to the identity. The user interface and permissions are loaded through the system user that is dynamically assigned to the logged in identity.</p> <p>Data modifications are attributed to the current user account.</p>

In the Designer, modify the following configuration parameters to implement the authentication module.

Table 30: Configuration parameters for the authentication module

Configuration parameter	Meaning
TargetSystem LDAP Authentication	Allows configuration of the LDAP authentication module.

Configuration parameter	Meaning
TargetSystem LDAP Authentication Authentication	<p>Authentication mechanism. Permitted values are Secure, Encryption, SecureSocketsLayer, ReadonlyServer, Anonymous, FastBind, Signing, Sealing, Delegation, and ServerBind. The value can be combined with commas (,). For more information about authentication types, see the MSDN Library.</p> <p>Default: ServerBind</p>
TargetSystem LDAP Authentication Port	<p>Communications port on the server.</p> <p>Default: 389</p>
TargetSystem LDAP Authentication RootDN	<p>Pipe () delimited list of root domains to be used to find the user account for authentication.</p> <p>Syntax:</p> <p>DC=<MyDomain> DC=<MyOtherDomain></p> <p>Example:</p> <p>DC=Root1,DC=com DC=Root2,DC=de</p>
TargetSystem LDAP Authentication Server	<p>Name of the LDAP server.</p>
TargetSystem LDAP AuthenticationV2	<p>Allows configuration of the LDAP authentication module.</p>
TargetSystem LDAP AuthenticationV2 AcceptSelfSigned	<p>Specifies whether self-signed certificates are accepted.</p>
TargetSystem LDAP AuthenticationV2 Authentication	<p>Authentication method for logging in to LDAP. The following are permitted:</p> <ul style="list-style-type: none"> • Basic: Uses default authentication. • Negotiate: Uses Negotiate authentication from Microsoft. • Kerberos: Uses Kerberos authentication. • NTLM: Uses Windows NT Challenge/Response (NTLM) authentication. <p>Default: Basic</p> <p>For more information about authentication types, see the MSDN Library.</p>
TargetSystem LDAP AuthenticationV2	<p>Client timeout in seconds.</p>

Configuration parameter	Meaning
ClientTimeout	
TargetSystem LDAP AuthenticationV2 Port	Communications port on the server. Default: 389
TargetSystem LDAP AuthenticationV2 ProtocolVersion	Version of the LDAP protocol. The values 2 and 3 are permitted. Default: 3
TargetSystem LDAP AuthenticationV2 RootDN	Pipe () delimited list of root domains to be used to find the user account for authentication. Syntax: DC=<MyDomain> DC=<MyOtherDomain> Example: DC=Root1,DC=com DC=Root2,DC=de
TargetSystem LDAP AuthenticationV2 Security	Connection security. Permitted values are None , SSL and STARTTLS .
TargetSystem LDAP AuthenticationV2 Server	Name of the LDAP server.
TargetSystem LDAP AuthenticationV2 UseSealing	Specifies whether sealing is enabled.
TargetSystem LDAP AuthenticationV2 UseSigning	Specifies whether signing is enabled.
TargetSystem LDAP AuthenticationV2 VerifyServerCertificate	Specifies whether to check the server certificate when encrypting with SSL.

Related topics

- [Configuration data for system user dynamic authentication](#) on page 114

HTTP header

| **NOTE:** This authentication module is available if the Configuration Module is installed.

The authentication module supports authentication by web single sign-on solutions that work with a proxy-based architecture.

Credentials	Identity's central user account or personnel number.
Prerequisites	<ul style="list-style-type: none"> • The system user with permissions exists in the One Identity Manager database. • The identity exists in the One Identity Manager database. • The central user account or personnel number is entered in the identity's main data. • The system user is entered in the identity's main data.
Set as default	No
Single sign-on	Yes
Front-end login allowed	No
Web Portal login allowed	Yes
Remarks	<p>You must pass the user (in the form: <code>UserName =<user name of authenticated user></code>) in the HTTP header. The identity found in the One Identity Manager database has the central user account or personnel number that matches the given user name.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none"> • If this configuration parameter is set, the identity's main identity is used for authentication. • If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p> <p>The user interface and permissions are loaded through the system user that is directly assigned to the logged in identity. If a system user is not assigned to the identity, the system user from the SysConfig Logon DefaultUser configuration parameter is used.</p> <p>Changes to the data are assigned to the logged in identity.</p>

HTTP header (role-based)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

The authentication module supports authentication by web single sign-on solutions that work with a proxy-based architecture.

Credentials	Identity's central user account or personnel number.
Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• The central user account or personnel number is entered in the identity's main data.• The identity is assigned at least one application role.
Set as default	Yes
Single sign-on	Yes
Front-end login allowed	No
Web Portal login allowed	Yes
Remarks	<p>You must pass the user (in the form: UserName =<user name of authenticated user>) in the HTTP header. The identity found in the One Identity Manager database has the central user account or personnel number that matches the given user name.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.• If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p> <p>A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.</p> <p>Changes to the data are assigned to the logged in identity.</p>

OAuth 2.0/OpenID Connect

NOTE: This authentication module is available if the Identity Management Base Module is installed.

The authorization module supports the authorization code for OAuth 2.0 and OpenID Connect. For more information about the authorization code flow, see, for example, the [OAuth Specification](#) or the [OpenID Connect Specification](#).

This authentication module uses a Secure Token Service for logging in. This login procedure can be used with every Secure Token Service that can return an OAuth 2.0 token.

Credentials	Dependent on the authentication method of the secure token service.
Prerequisites	<ul style="list-style-type: none">• The system user with permissions exists in the One Identity Manager database.• The identity exists in the One Identity Manager database.• The system user is entered in the identity's main data.• The user account exists in the One Identity Manager database and the identity is entered in the user account's main data.
Set as default	No
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>One Identity Manager finds the identity assigned to the user account.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.• If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p> <p>The user interface and permissions are loaded through the system user that is directly assigned to the identity found.</p> <p>Data modifications are attributed to the current user account. To do this,</p>

the claim type whose value is used for labeling data changes must be declared.

NOTE: If the authentication module cannot find a matching user for the claim value, it searches for the claim value in permitted system users' credentials (DialogUser.AuthenticatorLogons). If an entry is found there, then that system user is logged in. To allocate the data changes, the values are used from the respective claims. If a matching user is found, the fallback cannot be used anymore.

Related topics

- [OAuth 2.0/OpenID Connect authentication](#) on page 120
- [Expiry of the OAuth 2.0/OpenID Connect authentication](#) on page 121

OAuth 2.0/OpenID Connect (role-based)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

The authorization module supports the authorization code for OAuth 2.0 and OpenID Connect. For more information about the authorization code flow, see, for example, the [OAuth Specification](#) or the [OpenID Connect Specification](#).

This authentication module uses a Secure Token Service for logging in. This login procedure can be used with every Secure Token Service that can return an OAuth 2.0 token.

Credentials	Dependent on the authentication method of the secure token service.
Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• The identity is assigned at least one application role.• The user account exists in the One Identity Manager database and the identity is entered in the user account's main data.
Set as default	No
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	One Identity Manager finds the identity assigned to the user account. If an identity has a main identity or several subidentities, the QER

Person | MasterIdentity | UseMasterForAuthentication

configuration parameter controls which identity is used for authentication.

- If this configuration parameter is set, the identity's main identity is used for authentication.
- If this configuration parameter is set, the identity's subidentity is used for authentication.

NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the **QER | Person | AllowLoginWithSecurityIncident** configuration parameter.

A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.

Data modifications are attributed to the current user account. To do this, the claim type whose value is used for labeling data changes must be declared.

NOTE: If the authentication module cannot find a matching user for the claim value, it searches for the claim value in permitted system users' credentials (DialogUser.AuthentifierLogons). If an entry is found there, then that system user is logged in. To allocate the data changes, the values are used from the respective claims. If a matching user is found, the fallback cannot be used anymore.

Related topics

- [OAuth 2.0/OpenID Connect authentication on page 120](#)
- [Expiry of the OAuth 2.0/OpenID Connect authentication on page 121](#)

Synchronization authentication module

NOTE: This authentication module is available if the Target System Synchronization Module is installed.

This authentication module integrates the default method for Synchronization Editor login.

Credentials	Login uses the sa system user.
Prerequisites	
Set as default	Yes
Single sign-on	No
Front-end login	No

allowed

Web Portal login No
allowed

Remarks You must not change the system user **sa**. The system user is
overwritten with each schema update.

Web agent authentication module

| NOTE: This authentication module is available if the Configuration Module is installed.

The authentication module integrates the default method for Web Designer login, to access the database before the first user login.

Credentials Login uses the **sa** system user.

Prerequisites

Set as default Yes

Single sign-on No

Front-end login No
allowed

Web Portal login No
allowed

Remarks You must not change the system user **sa**. The system user is
overwritten with each schema update.

Component authentication module

| NOTE: This authentication module is available if the Configuration Module is installed.

This authentication module integrates the default method for registering process components.

Credentials Login uses the **sa** system user.

Prerequisites

Set as default Yes

Single sign-on No

Front-end login No

allowed

Web Portal login No
allowed

Remarks You must not change the system user **sa**. The system user is overwritten with each schema update.

Crawler

| NOTE: This authentication module is available if the Configuration Module is installed.

The authentication module is used by the application server to compile search indexes for full text search over the database.

Credentials Login uses the **sa** system user.

Prerequisites

Set as default Yes

Single sign-on No

Front-end login No
allowed

Web Portal login No
allowed

Remarks You must not change the system user **sa**. The system user is overwritten with each schema update.

Password reset

| NOTE: This authentication module is available if the Identity Management Base Module is installed.

The authentication module is used for login to Password Reset Portal. The authentication module checks the passcode or the identity's answers to the password questions. In the case of login with an passcode, this information is deleted after a successful login.

Credentials Central user account and passcode.

- OR -

Central user account and answers to the password questions.

- OR -

Target system user account and passcode.

- OR -

Target system user account and answers to password questions.

Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• Using the central user account: The central user account is entered in the identity main data.• Using the target system user account: The user account exists in the One Identity Manager database and the identity is entered in the main data of the identity's user account.• The identity is not deactivated or has the certification status New.• The identity has an passcode or the questions and answers for the password prompt have been specified.
Set as default	No
Single sign-on	No
Front-end login allowed	No
Web Portal login allowed	No
Remarks	The application token for Password Reset Portal must be specified. You set the application token when installing Password Reset Portal. The application token is saved as a hash value in the database in the QER Person PasswordResetAuthenticator ApplicationToken parameter and stored encrypted in the web.config file. For more information about setting up the Password Reset Portal, see the <i>One Identity Manager Web Application Configuration Guide</i> .

In the Designer, modify the following configuration parameters so that target system accounts can be used for logging in. If the configuration parameters are not set, the identity's central user account is used.

Table 31: Configuration parameters for the authentication module

Configuration parameter	Meaning
QER Person PasswordResetAuthenticator SearchTable	Table in the One Identity Manager schema which stores the user information. The table must contain a foreign key with the name UIDPerson (or CCC_UID_Person) that references the Person table. Example: ADSAccount
QER Person PasswordResetAuthenticator SearchColumn	Pipe () delimited list of columns from the One Identity Manager table (SearchTable) used to search for the user name of the logged in user.

Configuration parameter	Meaning
	<p>Example: CN SamAccountName</p> <p>NOTE: The QBMSplittedLookup table can be used as a lookup table. SplittedElement can be used as a search column.</p>
QER Person PasswordResetAuthenticator EnabledBy	Pipe () delimited list of Boolean columns from the One Identity Manager table (SearchTable) enabled by the user account for the login.
QER Person PasswordResetAuthenticator DisabledBy	<p>Pipe () delimited list of Boolean columns from the One Identity Manager table (SearchTable) disabled by the user account for the login.</p> <p>Example: AccountDisabled</p>

Password reset (role-based)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

The authentication module is used for login to Password Reset Portal. The authentication module checks the passcode or the identity's answers to the password questions. In the case of login with an passcode, this information is deleted after a successful login.

Credentials	<p>Central user account and passcode.</p> <p>- OR -</p> <p>Central user account and answers to the password questions.</p> <p>- OR -</p> <p>Target system user account and passcode.</p> <p>- OR -</p> <p>Target system user account and answers to password questions.</p>
Prerequisites	<ul style="list-style-type: none"> • The identity exists in the One Identity Manager database. • Using the central user account: The central user account is entered in the identity main data. • Using the target system user account: The user account exists in the One Identity Manager database and the identity is entered in the main data of the identity's user account. • The identity is not deactivated or has the certification status New. • The identity has an passcode or the questions and answers for the password prompt have been specified.

- The identity is assigned at least one application role.

Set as default	Yes
Single sign-on	No
Front-end login allowed	No
Web Portal login allowed	No
Remarks	<p>The application token for Password Reset Portal must be specified. You set the application token when installing Password Reset Portal. The application token is saved as a hash value in the database in the QER Person PasswordResetAuthenticator ApplicationToken parameter and stored encrypted in the <code>web.config</code> file. For more information about configuring the Password Reset Portal, see the <i>One Identity Manager Web Application Configuration Guide</i>.</p> <p>A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.</p>

In the Designer, modify the following configuration parameters so that target system accounts can be used for logging in. If the configuration parameters are not set, the identity's central user account is used.

Table 32: Configuration parameters for the authentication module

Configuration parameter	Meaning
QER Person PasswordResetAuthenticator SearchTable	<p>Table in the One Identity Manager schema which stores the user information. The table must contain a foreign key with the name <code>UIDPerson</code> (or <code>CCC_UID_Person</code>) that references the <code>Person</code> table.</p> <p>Example: <code>ADSAccount</code></p>
QER Person PasswordResetAuthenticator SearchColumn	<p>Pipe () delimited list of columns from the One Identity Manager table (SearchTable) used to search for the user name of the logged in user.</p> <p>Example: <code>CN SamAccountName</code></p> <p>NOTE: The <code>QBMSplittedLookup</code> table can be used as a lookup table. <code>SplittedElement</code> can be used as a search column.</p>
QER Person PasswordResetAuthenticator EnabledBy	<p>Pipe () delimited list of Boolean columns from the One Identity Manager table (SearchTable) enabled by the user account for the login.</p>
QER Person PasswordResetAuthenticator DisabledBy	<p>Pipe () delimited list of Boolean columns from the One Identity Manager table (SearchTable) disabled by</p>

Configuration parameter	Meaning
DisabledBy	the user account for the login. Example: AccountDisabled

Decentralized identity

NOTE: This authentication module is available if the Identity Management Base Module is installed.

The authentication module can be used to log in using a decentralized identity.

Credentials	The identity's email address and decentralized identity.
Prerequisites	<ul style="list-style-type: none"> • The system user with permissions exists in the One Identity Manager database. • The identity exists in the One Identity Manager database. • The decentralized identity is entered in the identity main data. • The default email address or the contact email address is in the identity main data. • The system user is entered in the identity's main data.
Set as default	No
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>To identify the identity, the email address provided during login is verified against the default email address and the contact email address.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none"> • If this configuration parameter is set, the identity's main identity is used for authentication. • If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration</p>

parameter.

The user interface and permissions are loaded through the system user that is directly assigned to the logged in identity.

Changes to the data are assigned to the logged in identity.

Decentralized Identity (role-based)

NOTE: This authentication module is available if the Identity Management Base Module is installed.

The authentication module can be used to log in using a decentralized identity.

Credentials	The identity's email address and decentralized identity.
Prerequisites	<ul style="list-style-type: none">• The identity exists in the One Identity Manager database.• The decentralized identity is entered in the identity main data.• The default email address or the contact email address is in the identity main data.• The identity is assigned at least one application role.
Set as default	No
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	<p>To identify the identity, the email address provided during login is verified against the default email address and the contact email address.</p> <p>If an identity has a main identity or several subidentities, the QER Person MasterIdentity UseMasterForAuthentication configuration parameter controls which identity is used for authentication.</p> <ul style="list-style-type: none">• If this configuration parameter is set, the identity's main identity is used for authentication.• If this configuration parameter is set, the identity's subidentity is used for authentication. <p>NOTE: Identities that are classified as a security risk are no longer be able to log in to One Identity Manager. To allow login, set the QER Person AllowLoginWithSecurityIncident configuration parameter.</p>

A dynamic system user is determined from the identity's application roles. The user interface and the permissions are loaded through this system user.

Changes to the data are assigned to the logged in identity.

Editing authentication modules

Before you can use an authentication module for logging on, the following prerequisites must be fulfilled:

1. The authentication module must be enabled.
2. The authentication module must be assigned to the application.
3. The assignment of the authentication module to the application must be enabled.

This allows you to log in to the assigned application using this authentication module. Ensure that users found through the authentication module also have the required program function to use the program.

Detailed information about this topic

- [Enabling authentication modules](#) on page 106
- [Assigning authentication modules to applications](#) on page 107
- [Disabling or enabling authentication modules for applications](#) on page 108
- [Authentication module properties](#) on page 109
- [Initial data for authentication modules](#) on page 110
- [Configuration data for system user dynamic authentication](#) on page 114
- [One Identity Manager authentication modules](#) on page 72
- [Managing permissions to program functions](#) on page 66
- [Program functions for starting the One Identity Manager tools](#) on page 150

Enabling authentication modules

NOTE: After the initial schema installation, only the **System user** and **Component authenticator** authentication modules and the role-based authentication modules are enabled in One Identity Manager.

To use an authentication module for logging in, you must enable the authentication module. Perform the following steps to enable an authentication module.

To enable an authentication module

1. In the Designer, select the **Base data > Security settings > Authentication modules** category.
2. In the List Editor, select the authentication module.
3. In the **Properties** view, set the **Activated** property to **True**.
4. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Disabling or enabling authentication modules for applications](#) on page 108
- [Assigning authentication modules to applications](#) on page 107

Assigning authentication modules to applications

NOTE: Use non role-based authentication modules to log in to the Designer. Role-based authentication modules for logging in to the Designer are not supported.

If create custom authentication modules, assign them to the existing programs. In general, you do not need to change assignments of predefined authentication modules.

To assign an authentication module to an application

1. In the Designer, select the **Base data > Security settings > Authentication modules** category.
2. Select the **View > Select table relations** menu item and enable the DialogProductHasAuthentifier table.
3. In List Editor, select the authentication module.
4. Assign the application in the **Applications** edit view.
5. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Disabling or enabling authentication modules for applications](#) on page 108
- [Enabling authentication modules](#) on page 106

Disabling or enabling authentication modules for applications

NOTE: Use non role-based authentication modules to log in to the Designer. Role-based authentication modules for logging in to the Designer are not supported.

To use an authentication module for login, assignment of the authentication module to the application must be enabled.

To enable an authentication module for an application

1. In the Designer, select the **Base data > Security settings > Authentication modules** category.
2. Select the **View > Select table relations** menu item and enable the DialogProductHasAuthentifier table.
3. In List Editor, select the authentication module.
4. In the **Application** edit view, select the assigned application.
5. Disable the **Disable** option.
6. Select the **Database > Commit to database** and click **Save**.

To disable an authentication module for an application

1. In the Designer, select the **Base data > Security settings > Authentication modules** category.
2. Select the **View > Select table relations** menu item and enable the DialogProductHasAuthentifier table.
3. In List Editor, select the authentication module.
4. In the **Application** edit view, select the assigned application.
5. Enable the **Disable** option.
6. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Assigning authentication modules to applications](#) on page 107
- [Enabling authentication modules](#) on page 106

Authentication module properties

Table 33: Authentication module properties

Property	Meaning
Enabled	Specifies whether the authentication module can be used.
Display name	Display name for displaying the authentication module in the connection dialog of the administration tools.
Authentication module	Internal name of the authentication module.
Authentication type	Authentication module type. You can choose from Dynamic and Role based .
Processing status	The processing status is used for creating custom configuration packages.
Initial data	Initial data for logging in with this authentication module. Syntax: property1=value1;property2=value2 Example: User=<user name>;Password=<password>
Class	Authentication module class.
Assembly name	Name of the assembly file.
Sort order	Specify the order in which the modules are displayed in the login window.
Single sign-on	Specifies whether the authentication module may be authenticated without a password.
Select in front-end	Specifies whether the authentication module can be selected in the login window.

Related topics

- [Enabling authentication modules](#) on page 106
- [Assigning authentication modules to applications](#) on page 107
- [Disabling or enabling authentication modules for applications](#) on page 108
- [Initial data for authentication modules](#) on page 110

Initial data for authentication modules

Authentication data is formatted from the authentication module and its parameters and values. You can specify initial data for the parameters and their values. By default, the initial data is preset for each authentication process.

Syntax for authentication data:

Module=<authentication module>;<property1>=<value1>;<property2>=<value2>,...

Example:

Module=DialogUser;User=<user name>;Password=<password>

To set initial data for authentication modules

1. In the Designer, select the **Base data > Security settings > Authentication modules** category.
2. Select the authentication module and enter the data in **Initial data**.

Syntax:

property1=value1;property2=value2

Example:

User=<user name>;Password=<password>

Table 34: Authentication data for authentication modules

Authentication module	Display name	Parameters and meaning
DialogUser	System users	User: User name Password: The user's password
ADSAccount	Active Directory user account	No parameters required
DynamicADSAccount	Active Directory user account (dynamic)	Product: Usage. The system user is determined through the use case configuration data.
DynamicManualADS	Active Directory user account (manual input)	Product: Usage. The system user is determined through the use case configuration data. User: User name. The user's identity is determined from a predefined list of permitted Active Directory domains. In the TargetSystem ADS AuthenticationDomains configuration parameter, enter the permitted Active Directory domains.

Authentication module	Display name	Parameters and meaning
		Password: The user's password.
RoleBasedADSAccount	Active Directory user account (role-based)	No parameters required
RoleBasedManualADS	Active Directory user account (manual input/role-based)	User: User name. The user's identity is determined from a predefined list of permitted Active Directory domains. In the TargetSystem ADS AuthenticationDomains configuration parameter, enter the permitted Active Directory domains. Password: The user's password
Employee	Identity	User: Identity's central user account. Password: The user's password
DynamicPerson	Identity (dynamic)	Product: Usage. The system user is determined through the use case configuration data. User: User name. Password: The user's password
RoleBasedPerson	Identity (role-based)	User: User name. Password: The user's password.
HTTPHeader	HTTP header	Header: The HTTP header to use. KeyColumn: Comma delimited list of key columns in the Person table to be searched for user names. Default: CentralAccount, PersonnelNumber
RoleBasedHTTPHeader	HTTP header (role-based)	Header: The HTTP header to use. KeyColumn: Comma delimited list of key columns in the Person table to be searched for user names. Default: CentralAccount, PersonnelNumber
DynamicLdap	LDAP user account	User: User name. Default: CN, DistinguishedName,

Authentication module	Display name	Parameters and meaning
	(dynamic)	<p>UserID, UIDLDAP</p> <p>Password: The user's password</p>
RoleBasedLdap	LDAP user account (role-based)	<p>User: User name.</p> <p>Default: CN, DistinguishedName, UserID, UIDLDAP</p> <p>Password: The user's password</p>
RoleBasedGeneric	Generic single sign-on (role-based)	<p>SearchTable: Table in which to search for the user name of the logged in user. This table must contain a FK named UID_Person that points to the Person table.</p> <p>SearchColumn: Column from the SearchTable in which to search for the user name of the logged-in user.</p> <p>DisabledBy: Pipe () delimited list of Boolean columns which block a user account from logging in.</p> <p>EnabledBy: Pipe () delimited list of Boolean columns which release a user account for logging in.</p>
OAuth	OAuth 2.0/OpenID Connect	Dependent on the authentication method of the secure token service.
OAuthRoleBased	OAuth 2.0/OpenID Connect (role-based)	Dependent on the authentication method of the secure token service.
DialogUserAccountBased	Account based system user	No parameters required
QERAccount	User account	No parameters required
RoleBasedQERAccount	User account (role-based)	No parameters required
RoleBasedManualQERAccount	User account (manual input/role-based)	<p>User: User name. The user's identity is determined from a predefined list of permitted Active Directory domains. In the</p> <p>TargetSystem ADS AuthenticationDomains config-</p>

Authentication module	Display name	Parameters and meaning
		uration parameter, enter the permitted Active Directory domains. Password: The user's password
PasswordReset	Password reset	No parameters required
RoleBasedPasswordReset	Password reset (role-based)	No parameters required
DecentralizedId	Decentralized identity	Email: Default email address of the identity (Person.DefaultEmailAddress) or contact email address of the identity (Person.ContactEmail) Identifier: Decentralized identity of the identity (Person.DecentralizedIdentifier).
RoleBasedDecentralizedId	Decentralized Identity (role-based)	Email: Default email address of the identity (Person.DefaultEmailAddress) or contact email address of the identity (Person.ContactEmail) Identifier: Decentralized identity of the identity (Person.DecentralizedIdentifier).
Token		Internal authentication module in the application server for authentication using OAuth 2.0/OpenID Connect access tokens. For more information, see Setting up OAuth 2.0/OpenID Connect authentication for accessing the application server's REST API on page 131. URL: URL of the application server. ClientId: ID of the application on the identity provider. ClientSecret: Secret value for authentication at the token endpoint. TokenEndpoint: Uniform Resource Identifier (URL) of the token endpoint of the authorization server for returning the access token to the client for logging in.

Related topics

- [Configuration data for system user dynamic authentication](#) on page 114
- [One Identity Manager authentication modules](#) on page 72

Configuration data for system user dynamic authentication

In the case of dynamic authentication modules, the system user assigned to the identity is not used for the log in. The system user which is configured using the user interface special configuration data is taken instead.

TIP: For system users used for dynamic authentication modules, enable the **Disabled for direct login** option. This prevents direct login to One Identity Manager tools with these system users.

To specify configuration data

1. In the Designer, select the **Base data > Security settings > Programs** category.
2. Select the application and adjust the **Configuration data**.

Use XML syntax for entering the configuration data:

```
<DialogUserDetect>
  <Usermappings>
    <Usermapping
      DialogUser = "System user name"
      Selection = "Selection criterion"
    />
    <Usermapping
      DialogUser = "System user name"
    />
    ...
  </Usermappings>
</DialogUserDetect>
```

Enter the system user (DialogUser) in the Usermappings section. Specify which identity the given system user should use with the selection criterion (Selection). You are not obliged to enter a selection criterion for the assignment. The first system user that has the required assignment is used for the log in.

You can assign function groups to permissions groups on order to deal with complex permissions and user interface structures. The function groups allow you to map the functions an identity has in the company, for example, IT controller or branch manager.

Assign the function groups to the permissions groups. A function group can refer to several permissions groups and several function groups can refer to one permissions group.

If the FunctionGroupMapping section is in the configuration data, this is evaluated first and the system user that is found is used. The authentication module uses the system user that is the exact member of the permissions group found for the login. If none is found, the Usermapping section is evaluated.

```
<DialogUserDetect>
  <FunctionGroupMapping
    PersonToFunction = "View mapping identity to function group"
    FunctionToGroup = "View mapping function group to permissions group"
  />
  <Usermappings>
    <Usermapping
      DialogUser = "System user name"
      Selection = "Selection criterion"
    />
    ...
  </Usermappings>
</DialogUserDetect>
```

Related topics

- [Example of a simple system user assignment on page 115](#)
- [Example of a system user assignment using a selection criterion on page 116](#)
- [Example of a function group assignment on page 117](#)
- [Granting One Identity Manager schema permissions through permissions groups on page 39](#)

Example of a simple system user assignment

All identities should be able to see the user interface for an IT Shop in a web front-end, without taking table and column permissions into account.

To do this, set up a new application, for example **WebShop_Customer_Prd**, and adapt the configuration data as follows:

```
<DialogUserDetect>
  <Usermappings>
    <Usermapping
      DialogUser = "dlg_all"
```

```

    />
  </Usermappings>
</DialogUserDetect>

```

Create a new **WebShop_Customer_Grp** permissions group, which receives the user interface for the application comprising the menu items, interface forms and task definitions. The user interface could consist of the following menu items:

- Employee contact data
- Requesting a product
- Unsubscribing a product

Define a new **dlg_all** system user and include it in the **vi_DE-CentralPwd**, the **vi_DE-ITShopOrder**, and the **WebShop_Customer_Grp** permissions groups.

Related topics

- [Configuration data for system user dynamic authentication](#) on page 114
- [Example of a system user assignment using a selection criterion](#) on page 116
- [Example of a function group assignment](#) on page 117
- [Granting One Identity Manager schema permissions through permissions groups](#) on page 39

Example of a system user assignment using a selection criterion

The scenario described in the previous example is extended such that only the cost center manager can see an identity's leaving date. You need to add the input field **LeavingDate** to the contact data form to do this.

Permissions are used for controlling viewing and editing. Set up a new **dlg_kst** system user and include the system user in the **vi_DE-CentralPwd**, **vi_DE-ITShopOrder** and **WebShop_Customer_Grp** permissions groups. You should also give the system user read and write permissions to the `Person.Exitdate` column.

Extend the application configuration data in such a way that the cost center managers use the **dlg_kst** system user to log in. All other identities use the **dlg_all** system user to log in.

Change the configuration data as follows:

```

<DialogUserDetect>
  <Usermappings>
    <Usermapping
      DialogUser = "dlg_kst"
    >
  </Usermappings>
</DialogUserDetect>

```

```

        Selection = "select 1 where %uid% in (select uid_personhead from
        profitcenter)"
    />
    <Usermapping
        DialogUser = "dlg_all"
    />
</Usermappings>
</DialogUserDetect>

```

Related topics

- [Configuration data for system user dynamic authentication](#) on page 114
- [Example of a simple system user assignment](#) on page 115
- [Example of a function group assignment](#) on page 117
- [Granting One Identity Manager schema permissions through permissions groups](#) on page 39

Example of a function group assignment

To assign function groups to permissions groups you have to define two database views. The first database view shows the assignment of identities to function groups. The database view contains two columns, UID_Person and FunctionGroup.

Example:

```

create view custom_Person2Fu as
    select uid_personHead as UID_Person, 'Cost center manager' as FunctionGroup
    from Profitcenter
    where isnull(uid_personHead, '') > ' '
    union all
    select uid_personHead, 'Department manager' as FunctionGroup
    from Department
    where isnull(uid_personHead, '') > ' '

```

The second database view assigns function groups to permissions groups. This database view contains two columns, FunctionGroup and DialogGroup.

Example:

```

create view custom_Fu2D as
    select 'Cost center manager' as FunctionGroup, '<UID_Custom_Dialoggroup_
    ChefP>' as DialogGroup

```

```
union all select 'Department manager', '<UID_Custom_Dialoggroup_ChefD>' as
DialogGroup
```

Set up role-based permissions groups with the required permissions.

TIP: A role-based permissions group can inherit from a non role-based permissions group. This allows you to build up an inheritance hierarchy to making it easier to grant permissions.

Change the configuration data for assigning function groups to permissions groups as follows:

```
<DialogUserDetect>
  <FunctionGroupMapping
    PersonToFunction = "custom_Person2Fu"
    FunctionToGroup = "custom_Fu2D"
  />
</DialogUserDetect>
```

Related topics

- [Configuration data for system user dynamic authentication](#) on page 114
- [Example of a simple system user assignment](#) on page 115
- [Example of a system user assignment using a selection criterion](#) on page 116
- [Granting One Identity Manager schema permissions through permissions groups](#) on page 39

Checking authentication

When a user logs in, a validity check is run. Use the settings to configure additional options.

- 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 with next action on the connection after a fixed interval of 20 minutes.

You can adjust the interval in the **Common | Authentication | CheckInterval** configuration parameter. In the Designer, edit the configuration parameter.

- The number of session that a user can open within a short time is limited to 10 session a minute.

If this number is exceeded, the user is sent an error message.

You have logged in too often in the last minute. Please wait a moment before you log in again.

This check is done for each front-end if the login is local. If the login is on the application server, it is checked for each application server.

You can modify the number of sessions in the **Common | Authentication | SessionsPerUserAndMinute** configuration parameter. In the Designer, edit the configuration parameter.

- Use the **QBM | AppServer | SessionTimeout** configuration parameter to add the timeout in hours, after which inactive application server sessions are closed. The default value is **24** hours. In the Designer, edit the configuration parameter.

OAuth 2.0/OpenID Connect authentication

The **OAuth2.0/OpenID Connect** and **OAuth2.0/OpenID Connect (role-based)** authentication modules support the authorization code flow for OAuth 2.0 and OpenID Connect. For more information about the authorization code flow, see, for example, the [OAuth Specification](#) or the [OpenID Connect Specification](#).

To use OAuth2.0/OpenID Connect authentication

- In the Designer, create the identity provider and the OAuth2.0/OpenID Connect applications for the identity provider. A wizard is available in the Designer to assist in this process.
- Assign the OAuth2.0/OpenID Connect application to the web applications.

Related topics

- [Expiry of the OAuth 2.0/OpenID Connect authentication on page 121](#)
- [Creating the OAuth 2.0/OpenID Connect configuration on page 122](#)
- [Assigning OAuth 2.0/OpenID Connect configuration to web applications on page 127](#)
- [Specifying enabled and disabled columns for logging in on page 129](#)
- [OAuth 2.0/OpenID Connect on page 96](#)
- [OAuth 2.0/OpenID Connect \(role-based\) on page 97](#)
- [Logging information about OAuth 2.0/OpenID Connect authentication on page 130](#)
- [Setting up OAuth 2.0/OpenID Connect authentication for accessing the application server's REST API on page 131](#)
- [OAuth 2.0/OpenID Connect authentication on the API Server on page 134](#)

Expiry of the OAuth 2.0/OpenID Connect authentication

The web application (or client application) requests the authorization code at the authorization endpoint. The login endpoint is used to call an advanced login window, which serves to determine the authorization code. The authentication module requires an access token from the token endpoint and the certificate is required to check the security token.

In the process, an attempt is made to find the certificate from the web application configuration. If this is not possible, the settings of the identity provider are used. To find the certificate for testing the token, the certificate stores are queries in the following order:

1. Configuration of the OAuth 2.0/OpenID Connect application (QBMSecurityClient table)
 - a. Certificate text (QBMSecurityClient.CertificateText).
 - b. Subject or thumbprint from the local memory (QBMSecurityClient.CertificateSubject and QBMSecurityClient.CertificateThumbPrint).
 - c. Certificate endpoint (QBMSecurityClient.CertificateEndpoint).
In addition, the subject or thumbprint is used to check certificates from the server if they are specified and do not exist locally on the server.
2. Configuration of the identity provider (QBMSecurityProvider table)
 - a. Certificate text ((QBMSecurityProvider.CertificateText).
 - b. Subject or thumbprint from the local memory (QBMSecurityProvider.CertificateSubject and QBMSecurityProvider.CertificateThumbPrint).
 - c. Certificate endpoint (QBMSecurityProvider.CertificateEndpoint)).
In addition, the subject or thumbprint is used to check certificates from the server if they are specified and do not exist locally on the server.
 - d. JSON-Web-Key endpoint (QBMSecurityProvider.JsonWebKeyEndpoint).

To identify the user account, the system determines which claim type is used to find the user information and which information from the One Identity Manager schema is used to find the user account.

Authentication through OpenID is built on OAuth 2.0. The OpenID Connect authentication uses the same mechanisms, but makes the claims available either in an ID token or with a UserInfo endpoint. Other configuration settings are required for using OpenID Connect. If the **Scope** contains the **openid** value, the authentication module uses OpenID Connect for authentication.

Related topics

- [Creating the OAuth 2.0/OpenID Connect configuration](#) on page 122
- [OAuth 2.0/OpenID Connect](#) on page 96
- [OAuth 2.0/OpenID Connect \(role-based\)](#) on page 97

Creating the OAuth 2.0/OpenID Connect configuration

To create an OAuth 2.0/OpenID Connect configuration

1. In the Designer, select the **Base data > Security settings > OAuth 2.0/OpenID Connect configuration** category.
2. Select the **Create a new identity provider** task.
3. On the start page of the wizard, click **Next**.
4. On the **New identity provider** page, enter the display name for the configuration and a description.
5. Click **Next**.
6. On the **Automatic configuration discovery** page, you define how you want to enter the information about the identity provider.
 - If the configuration data can be determined automatically by OpenID Connect Discovery:
 1. Select **Automatic configuration data discovery**.
 2. Enter the address (URL) for automatic determination of the configuration data in the input field, or select an example address through the selection menu.
 3. Click **Run**.
 4. The configuration data is determined and a dialog window is displayed. To accept the configuration data, click **OK**.
 - If you want to create the configuration data from a template:
 1. Select **Create from template file**.
 2. Click **Select** and choose the XML file.

For the One Identity Redistributable STS (RSTS), the file is pre-configured. You can find the RSTS_Template.xml in the One Identity Manager installation directory.
 3. Click **Open**.
 - If you do not want to determined the configuration data automatically, select

Manual data input.

Enter the configuration data on the next page of the wizard.

7. Click **Next**.
8. On the **Configuration data** page, enter the general information for the database user.

NOTE: If you selected automatic determination of configuration data, some of the information is already completed.

Table 35: General configuration data for the identity provider

Property	Description
Login endpoint	Uniform Resource Locator (URL) of the Secure Token Service login page. Example: http://localhost/rsts/login
Logout endpoint	URL of the log-out endpoint Example: http://localhost/rsts/login?wa=wsignout1.0
Token endpoint	Uniform Resource Identifier (URL) of the token endpoint of the authorization server for returning the access token to the client for logging in. Example: https://localhost/rsts/oauth2/token
Issuer	Uniform Resource Identifier (URI) of the certificate issuer for verifying the security token. Example: urn:STS/identity
Scope	Protocol for authentication. If the value is openid , OpenID Connect is used for authentication, otherwise OAuth 2.0 is used.
UserInfo endpoint	URL of the OpenID Connect UserInfo endpoint.
No ID token check	Specifies whether a check is made of the ID token. If the option is enabled, the ID token is not checked. The option can only be enabled for a scope containing the value openid and a populated UserInfo endpoint.
Self-signed certificates allowed	Specifies whether self-signed certificates are allowed for connecting to the token endpoint and UserInfo endpoint.
Shared Secret	Shared-Secret value used for authentication at the token endpoint. If all applications of the identity provider use the same Shared Secret, enter the value here. If the applications use different Shared Secrets, enter the Shared Secret values when creating the applications.

Property	Description
Requested authentication context class reference values	Space-delimited string specifying the acr values that the authorization server ought to use to process this authentication request, with the values appearing in order of preference.

9. Click **Next**.
10. On the **Configure certificates** page, enter the information for the identity provider's certificate. If all applications use the same certificate, enter the information here. If the applications use different certificate settings, enter the information when creating the application.
NOTE: If you selected automatic determination of configuration data, some of the information is already completed.

Table 36: Information about the identity provider certificate

Property	Description
Certificate endpoint	Uniform Resource Locator (URL) of the certificate end point on the authorization server. Example: https://localhost/RSTS/SigningCertificate
Subject of the certificate	Subject of the certificate used for verification. The subject or thumbprint must be set.
Thumbprint	Thumbprint of the certificate used to verify the security token.
JSON-Web-Key endpoint	URL of the JSON web key endpoint providing the token signing keys.
Certificate	Character string of the certificate content. It is used if no certificate is configured.

11. Click **Next**.
12. On the **Search rule for user information** page, you define how the login information is determined between the identity provider and the One Identity Manager database.

Table 37: Determining the login information

Property	Description
Value for the search	Full name of the claim type from which the login credentials are determined on the identity provider.

Property	Description
	<p>Example: name of an entity</p> <p><code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier</code></p> <p>If you have determined the configuration data automatically, select a value from the list.</p>
Column to search	<p>Table and column in the One Identity Manager database in which the user information is stored. The table must contain a foreign key with the name <code>UID_Person</code>, which points to the <code>Person</code> table.</p> <p>Example: <code>ADSAccount.ObjectGUID</code></p>
User name value	<p>Full name of the claim type from which the user name is determined on the identity provider. The user name is used, for example, to identify data changes in One Identity Manager (<code>XUserInserted</code> and <code>XUserUpdated</code> columns).</p> <p>Example: User Principle Name (UPN)</p> <p><code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn</code></p> <p>If you have determined the configuration data automatically, select a value from the list.</p>
Value to check	<p>Name of the claim type to be additionally checked. The claim type must appear under exactly this name in the token. The check ensures that only those people can log in whose token contains exactly the comparison value in the specified claim type.</p>
Comparison value	<p>Fixed value of the claim type specified under Value to check, against which is checked.</p>

13. Click **Next**.

14. On the **Create OAuth 2.0/OpenID Connect applications** page, enter the application information for the identity provider.

- a. Click  next to the **Applications** field.

To connect using RSTS, select **RSTS client**. Some of the information about the **RSTS client** application is already predefined.

- b. On the **General** tab, enter the general information for the application.

Table 38: General information about the application

Property	Description
Display name	Display name of the application.

Property	Description
Description	Text field for additional explanation.
Client ID	ID of the application on the identity provider. For client applications, enable the Default option. Example: urn:OneIdentityManager/Web
Shared Secret	Application-specific Shared Secret value used for authentication at the token endpoint.
Resource to request	URN of the resource to be requested, for example for ADFS. Only required if the identity provider requires this value.
Redirect URL	Forwarding address for redirection of applications. Example: urn:InstalledApplication
Send post logout redirect URI	Specifies the behavior of the client after logging off from the application. Permitted values are Send post logout redirect URI (default), Do not send a redirect URI , and Send a specific redirect URI .
Post logout redirect URI	URI sent after logging off from the application.
Default	Specifies whether this is a standard application for client applications.

- c. On the **Certificate** tab, enter the information for the application certificate.

Table 39: Information about the application certificate

Property	Description
Certificate endpoint	Uniform Resource Locator (URL) of the certificate end point on the authorization server. Example: https://localhost/RSTS/SigningCertificate
Thumbprint	Thumbprint of the certificate used to verify the security token.
Subject of the certificate	Subject of the certificate used for verification. The subject or thumbprint must be set.
Certificate	Content of the certificate. It is used if no certificate is configured.

- d. On the **Authentication** tab, enter the following information

Table 40: Information about the application certificate

Property	Description
Authentication method	Authentication method at the token endpoint. Permitted values are: <ul style="list-style-type: none">• client_secret_basic (default value): HTTP basic authentication method. The Shared Secret is transferred in the HTTP header.• client_secret_post: The Shared Secret is transferred in the client_secret value of the POST-Body.• none: No authentication at the token endpoint.• client_secret_jwt: The Shared Secret is transferred as a JSON web token (JWT).• private_key_jwt: The Shared Secret is transferred as JWT. In addition, encryption is carried out with the private key.
Token endpoint certificate	Hexadecimal thumbprint of the certificate for validating the token.
Requested authentication context class reference values	Space-delimited string specifying the acr values that the authorization server ought to use to process this authentication request, with the values appearing in order of preference. If no reference values are defined here, the reference values of the identity provider are used.

15. To create the identity provider and the application in the One Identity Manager database, click **Next**.

16. Click **Finish** to complete the wizard.

Related topics

- [Installing One Identity Redistributable Secure Token Server](#) on page 145
- [Multi-factor authentication with One Identity Defender](#) on page 138

Assigning OAuth 2.0/OpenID Connect configuration to web applications

To use the **OAuth2.0/OpenID Connect** and **OAuth2.0/OpenID Connect (role-based)** authentication modules in One Identity Manager web applications, assign the

OAuth2.0/OpenID Connect application to the web application.

To assign an OAuth2.0/OpenID Connect application to a web application

1. In the Designer, select the **Base data > Security settings > Web server configurations** category.
2. In List Editor, select the web application.
3. In the **Properties** edit view, assign the application in the **OAuth2.0/OpenID Connect application** selection list.
4. Select the **Database > Commit to database** and click **Save**.

TIP: For some web applications, for example the Web Portal, you can customize the OAuth2.0/OpenID Connect configuration in the configuration file (web.config). For detailed information about configuring the Web Portal, see the *One Identity Manager Installation Guide*.

Displaying the configuration of the identity provider and the OAuth 2.0/OpenID Connect applications

To display the configuration of an identity provider

1. In the Designer, select the **Base data > Security settings > OAuth 2.0/OpenID Connect configuration** category.
2. In List Editor, select the identity provider. The configuration data is displayed on the following tabs in the edit view.
 - **General:** Displays the general configuration data of the identity provider.
 - **Certificate:** Shows the information about the identity provider certificate.
 - **Applications:** Displays the configuration of the OAuth 2.0/OpenID Connect applications.
 - **Columns for enabling:** Displays the table and the columns that identify a user account as activated.
 - **Columns for disabling:** Displays the table and the columns that identify a user account as deactivated.

To display the configuration of an OAuth 2.0/OpenID Connect application

1. In the Designer, select the **Base data > Security settings > OAuth 2.0/OpenID Connect configuration** category.
2. In List Editor, select the identity provider.
3. In the edit view, select the **Applications** tab.

4. To display the configuration of an application, select the OAuth 2.0/OpenID Connect application in the **Application** view.

NOTE:

Click on **Add** to add a new OAuth 2.0/OpenID Connect application to the configuration of the identity provider.

Click on **Remove** to remove an OAuth 2.0/OpenID Connect application that is no longer required from the configuration of the identity provider.

Related topics

- [Creating the OAuth 2.0/OpenID Connect configuration](#) on page 122
- [Specifying enabled and disabled columns for logging in](#) on page 129

Specifying enabled and disabled columns for logging in

In the determination of the user account for the OAuth 2.0/OpenID Connect authentication, the system checks whether the user account is enabled or disabled. You define which columns can mark a user account as enabled or disabled.

Note:

- Only the columns of the table that you selected in the OAuth 2.0/OpenID Connect configuration of the identity provider in the **Column to search** are displayed.
- A column can either be used as an enabled or a disabled column.
- You can specify just enabled columns or just disabled columns, or a combination of enabled and disabled columns.

Example:

A search column references the ADSAccount table.

Case a) Only enabled Active Directory user accounts are allowed to login.

- Select ADSAccount.AccountDisabled as the disabled column.

If the ADSAccount.AccountDisabled column of the user account is set, login is not permitted.

Case b) Only privileged Active Directory user accounts are allowed to login.

- Select `ADSAccount.IsPrivilegedAccount` as the enabled column.

If the `ADSAccount.IsPrivilegedAccount` column of the user account is set, login is permitted.

Case c) Only enabled, privileged Active Directory user accounts are allowed to login.

- Select `ADSAccount.IsPrivilegedAccount` as the enabled column and `ADSAccount.AccountDisabled` as the disabled column.

If the `ADSAccount.IsPrivilegedAccount` column of the user account is set and the `ADSAccount.AccountDisabled` column of the user account is not set, login is permitted.

To define which columns can enable a user account for login

1. In the Designer, select the **Base data > Security settings > OAuth 2.0/OpenID Connect configuration** category.
2. In the List Editor, select the configuration.
3. In the edit view, select the **Columns for enabling** tab.
4. In the **Add assignment** view, assign the columns that enable the user account for login.
5. Select the **Database > Commit to database** and click **Save**.

To define which columns can disable a user account for login

1. In the Designer, select the **Base data > Security settings > OAuth 2.0/OpenID Connect configuration** category.
2. In the List Editor, select the configuration.
3. Select the **Columns for disabling** tab in the edit view.
4. In the **Add assignment** view, assign the columns that disable the user account for login.
5. Select the **Database > Commit to database** and click **Save**.

Logging information about OAuth 2.0/OpenID Connect authentication

To support troubleshooting in OAuth 2.0/OpenID Connect authentication you can log personal login data, such as information about tokens or issuers. The log is written to the object log file (`<appName>_object.log`) of the respective One Identity Manager component.

To log authentication data

- In the Designer, set the **QBM | DebugMode | OAuth2 | LogPersonalInfoOnException** configuration parameter.

Setting up OAuth 2.0/OpenID Connect authentication for accessing the application server's REST API

The One Identity Manager REST API is an integral part of the application server. To use OAuth 2.0/OpenID Connect authentication for accessing the application server's REST API, there is support for the **OAuth2.0/OpenID Connect** and **OAuth2.0/OpenID Connect (role-based)** authentication modules.

Authentication is done using the access token provided. The first time a request is made with a new access token, a session is established with that token and the authentication module. Further accesses with the same token use the same session. The validity period of the token is checked in the process.

For more information about the One Identity Manager REST API, see *One Identity Manager REST API Reference Guide*.

Related topics

- [Expiry of the OAuth 2.0/OpenID Connect authentication on page 121](#)
- [Setting up OAuth 2.0/OpenID Connect authentication the on application server on page 131](#)
- [Authentication module for using OAuth 2.0/OpenID Connect for authentication access to the REST API on page 132](#)
- [Authenticating external applications on the application server using OAuth 2.0/OpenID Connect on page 133](#)

Setting up OAuth 2.0/OpenID Connect authentication the on application server

NOTE: To access the REST API in the application server, users need the **AppServer_API** program function.

To set up authentication for the REST API using OAuth 2.0/OpenID Connect

- In the Designer, set the **QBM | AppServer | AccessTokenAuth** configuration parameter.
- In the Designer, set the respective authentication module either **OAuth 2.0/OpenID Connect** or **OAuth 2.0/OpenID Connect (role-based)**.
- If the **OAuth 2.0/OpenID Connect (role-based)** authentication module is used, set the **QBM | AppServer | AccessTokenAuth | RoleBased** configuration parameter as well.
- In the Designer, create the OAuth 2.0/OpenID Connect configuration and assign the configuration to the web application for the application server.
- The URL for the application server must be declared.

When the application server is installed, an entry for the web application is created with the URL in the QBMWebApplication table. Check whether the URL (BaseURL column) is entered.

To display a web application's settings

1. In the Designer, select the **Base data > Security settings > Web server configurations** category.
2. In List Editor, select the web application.

Related topics

- [Creating the OAuth 2.0/OpenID Connect configuration](#) on page 122
- [Assigning OAuth 2.0/OpenID Connect configuration to web applications](#) on page 127
- [OAuth 2.0/OpenID Connect](#) on page 96
- [OAuth 2.0/OpenID Connect \(role-based\)](#) on page 97
- [Enabling authentication modules](#) on page 106

Authentication module for using OAuth 2.0/OpenID Connect for authentication access to the REST API

An authentication module is provided within the application server to authenticate using access tokens. The application server client uses the information from the authentication module to determine the access token for logging in on the server side.

For example, the authentication module can be used for Job servers that do not have a direct connection to the database but work against an application server.

To use the authentication module, ensure that authentication for accessing the REST API is set up using OAuth 2.0/OpenID Connect.

NOTE: If authentication is by access token, other authentication modules are excluded from use and the application server returns an error.

Authentication data for establishing a connection through the application server's REST API.

Module=Token;Url=<URL of the application server>;ClientId=<client-ID>;ClientSecret=<secret>;TokenEndpoint=<token endpoint>.

With the following parameters:

- URL: URL of the application server
- ClientId: Client ID for authentication at the token endpoint.
- ClientSecret: Secret value for authentication at the token endpoint.
- TokenEndpoint: URL of the token endpoint.

For more information about providing connection and authentication data to the application server for Job servers, see the *One Identity Manager Configuration Guide*.

Related topics

- [Setting up OAuth 2.0/OpenID Connect authentication the on application server](#) on page 131

Authenticating external applications on the application server using OAuth 2.0/OpenID Connect

To access the REST API in the application server through external applications, authentication is supported by the **OAuth2.0/OpenID Connect** and **OAuth2.0/OpenID Connect (role-based)** authentication modules. Ensure that authentication for the REST API is set up through OAuth 2.0/OpenID Connect.

To authenticate an external application using Oauth 2.0/Openid Connect in One Identity Manager

1. Log in to the external identity provider, for example with Redistributable STS (RSTS), and get the access token.
2. Ensure that the token is passed as the bearer token in the authentication header of all queries.

NOTE: The session must be handled by a bearer token when logging in using a session cookie. Clients accessing the REST API using the bearer token must therefore keep the cookie assigned during the first access and send it with subsequent accesses. Otherwise, a new session is established for each access, which costs a lot of resources.

Related topics

- [Setting up OAuth 2.0/OpenID Connect authentication the on application server on page 131](#)
- [OAuth 2.0/OpenID Connect on page 96](#)
- [OAuth 2.0/OpenID Connect \(role-based\) on page 97](#)

OAuth 2.0/OpenID Connect authentication on the API Server

To use OAuth 2.0/OpenID Connect authentication for accessing the API Server's REST API, there is support for the **OAuth2.0/OpenID Connect** and **OAuth2.0/OpenID Connect (role-based)** authentication modules.

Authentication is done using the access token provided. The first time a request is made with a new access token, a session is established with that token and the authentication module. Further accesses with the same token use the same session. The validity period of the token is checked in the process.

Related topics

- [Expiry of the OAuth 2.0/OpenID Connect authentication on page 121](#)
- [Setting up OAuth 2.0/OpenID Connect authentication on the API Server on page 134](#)
- [Authenticating external applications on the API Server using OAuth 2.0/OpenID Connect on page 135](#)

Setting up OAuth 2.0/OpenID Connect authentication on the API Server

To set up authentication using OAuth 2.0/OpenID Connect

- In the Designer, set the **QBM | AppServer | AccessTokenAuth** configuration parameter.
- In the Designer, set the respective authentication module either **OAuth 2.0/OpenID Connect** or **OAuth 2.0/OpenID Connect (role-based)**.
- If the **OAuth 2.0/OpenID Connect (role-based)** authentication module is used, set the **QBM | AppServer | AccessTokenAuth | RoleBased** configuration parameter as well.
- In the Designer, create the OAuth 2.0/OpenID Connect configuration and assign the configuration to the web application for the API Server.

- The URL for the API Server must be declared.

When the API Server is installed, an entry for the web application is created with the URL in the QBMWebApplication table. Check whether the URL (BaseURL column) is entered.

To display a web application's settings

1. In the Designer, select the **Base data > Security settings > Web server configurations** category.
2. In List Editor, select the web application.

Related topics

- [Creating the OAuth 2.0/OpenID Connect configuration](#) on page 122
- [Assigning OAuth 2.0/OpenID Connect configuration to web applications](#) on page 127
- [OAuth 2.0/OpenID Connect](#) on page 96
- [OAuth 2.0/OpenID Connect \(role-based\)](#) on page 97
- [Enabling authentication modules](#) on page 106

Authenticating external applications on the API Server using OAuth 2.0/OpenID Connect

To access the REST API in the API Server through external applications, authentication is supported by the **OAuth2.0/OpenID Connect** and **OAuth2.0/OpenID Connect (role-based)** authentication modules. Ensure that authentication for the REST API is set up through OAuth 2.0/OpenID Connect.

To authenticate an external application using OAuth 2.0/Openid Connect in One Identity Manager

1. Log in to the external identity provider, for example with Redistributable STS (RSTS), and get the access token.
2. Ensure that the token is passed as the bearer token in the authentication header of all queries.

NOTE: The session must be handled by a bearer token when logging in using a session cookie. Clients accessing the REST API using the bearer token must therefore keep the cookie assigned during the first access and send it with subsequent accesses. Otherwise, a new session is established for each access, which costs a lot of resources.

Related topics

- [OAuth 2.0/OpenID Connect authentication on the API Server](#) on page 134
- [OAuth 2.0/OpenID Connect](#) on page 96
- [OAuth 2.0/OpenID Connect \(role-based\)](#) on page 97

Multi-factor authentication in One Identity Manager

One Identity Defender can be used for multi-factor authentication on One Identity Manager tools and the Web Portal . For more information, see [Multi-factor authentication with One Identity Defender](#) on page 138.

You can set up multi-factor authentication with OneLogin for attestations and request approvals. For more information, see [Multi-factor authentication with OneLogin](#) on page 137.

Multi-factor authentication with OneLogin

You can set up multi-factor authentication with OneLogin for specific security-critical actions in One Identity Manager. You can use these, for example, for attestation or when approving requests in the Web Portal. Each identity that wants to use this functionality, must be linked to a OneLogin user account.

Prerequisite

In OneLogin:

- At least one authentication method is configured on all user accounts that are going to use multi-factor authentication.

In One Identity Manager:

- The OneLogin Module is installed.

To use multi-factor authentication for attestations or requests

1. Set up synchronization with a OneLogin domain and start the synchronization.
2. Link identities to their OneLogin user accounts.

3. Configure the API Server and the Web Portal for using OneLogin multi-factor authentication.
4. Set up multi-factor authentication for attestations and requests in the IT Shop.

For more information, see the following guides:

Theme	Guide
Set up and start synchronization of a OneLogin domain.	One Identity Manager Administration Guide for Integration with OneLogin Cloud Directory
Multi-factor authentication configuration in the web application	One Identity Manager Web Application Configuration Guide
Preparing the IT Shop for multi-factor authentication	One Identity Manager IT Shop Administration Guide
Setting up multi-factor authentication for attestation	One Identity Manager Attestation Administration Guide
Requesting products requiring multi-factor authentication	One Identity Manager Web Portal User Guide
Approving requests with multi-factor authentication	
Attestation with multi-factor authentication	

Multi-factor authentication with One Identity Defender

One Identity Defender can be used for multi-factor authentication on One Identity Manager tools and the Web Portal . A Redistributable STS (RSTS) is set up to provide Active Directory authentication over a RADIUS server.

Prerequisite

- One Identity Defender is installed and set up.

To set up multi-factor authentication using Defender

1. Install the RSTS.

In the Installation Wizard on the **Installation Settings** page, enter the signing certificate, URL, and configuration password for the RSTS administration interface. For test or demonstration environments, you can use the **Redistributable STS Demo** signing certificate.

2. Configure the RSTS.
3. Set up the OAuth 2.0/OpenID Connect configuration.
In doing so, you create a new identity provider. You will need this identity provider for configuring authentication with OAuth 2.0/OpenID Connect.
4. Configure authentication with OAuth 2.0/OpenID Connect for the Web Portal.
5. Configure authentication with OAuth 2.0/OpenID Connect for the One Identity Manager administration tools.
6. Test the access to the Web Portal.
 - After entering the URL of the Web Portals in your web browser, you should be redirected to the RSTS login page.
 - After logging in with user name and password, you are prompted to enter your Defender Token.

If both authentications were successful, you can work with the Web Portal.

7. Test access to the One Identity Manager administration tools.
 - Start an administration tool, for example, the Launchpad, and select the **OAuth 2.0/OpenID Connect** authentication method.
 - After logging in with user name and password, you are prompted to enter your Defender Token.

If both authentications were successful, you can work with the administration tool.


Detailed information about this topic

- [Installing One Identity Redistributable Secure Token Server](#) on page 145
- [Configuring RSTS for multi-factor authentication](#) on page 139
- [Creating the OAuth 2.0/OpenID Connect configuration](#) on page 122
- [Configuring authentication with OAuth 2.0/OpenID Connect in the Web Portal](#) on page 140
- [Configuring authentication with OAuth 2.0/OpenID Connect](#) on page 141

Configuring RSTS for multi-factor authentication

To configure multi-factor authentication using a RADIUS server on the RSTS

1. Start a web browser and open the URL of the RSTS administration interface.
`https://<webapplication>/RSTS/admin`
Use the configuration password assigned during installation to log in.
2. On the home page, click **Authentication providers**.

3. On the **Authentication Providers** page, select the **Default Active Directory** default provider and click  **Edit**.
4. On the **Edit** page, select the **Authentication provider** tab and edit the following settings.
 - **Directory Type > Active Directory**: enabled
 - **Connection Information > Use Current Domain**: enabled
5. Select the **Two Factor Authentication** tab and edit the settings for your Defender Security Server.
 - **Two Factor Authentication Settings > RADIUS**: enabled
 - **Server, Port, Shared Secret**, and **Username Attribute**: Connection data for the RADIUS server.
 - (Optional) **Connection Information > Pre-authenticate For ChallengeResponse**: Uses the response text of the defender, instead of the default RADIUS response text.
6. Switch to the home page and select **Applications**.
7. On the **Applications** page, click **Add Application**.
8. On the **Edit** page, select the **General Settings** tab and edit the following settings.
 - **Application Name, Authentication Provider, Realm/Client_ID/Issuer, Redirect Url**

The redirect URL for the Web Portal (**Redirect Url**) is formed as follows:
`https://<Server>/<Application Name>/`
9. Select the **Certificates** tab and under **Signing Certificate (Required)** activate the signing certificate that you specified when installing the RSTS.

For more information, see [Multi-factor authentication with One Identity Defender](#) on page 138.
10. Click **Finish**.


Related topics

- [Installing One Identity Redistributable Secure Token Server](#) on page 145

Configuring authentication with OAuth 2.0/OpenID Connect in the Web Portal

To configure authentication with OAuth 2.0/OpenID Connect

1. Start the Web Designer.
2. Click the **View > Home page** menu item.
3. On the home page, click **Select web application** and select the web application.

4. Click  **Edit web application settings**.
5. In the **Edit web application settings** dialog, edit the web application settings.
 - **Authentication module:** Select **OAuth 2.0/OpenID Connect (role-based)**.
 - **OAuth 2.0/OpenID Connect configuration:** Select the newly created identity provider.
 - **Client ID for OAuth 2.0 authentication:** Select the client ID that you specified when you configured RSTS.
 - **Fingerprint of the OAuth 2.0 certificate:** Specify the fingerprint of the signing certificate you selected when configuring the RSTS.
6. Save the changes.

Related topics

- [Multi-factor authentication with One Identity Defender](#) on page 138
- [Configuring RSTS for multi-factor authentication](#) on page 139
- [Configuring authentication with OAuth 2.0/OpenID Connect](#) on page 141
- [Creating the OAuth 2.0/OpenID Connect configuration](#) on page 122

Configuring authentication with OAuth 2.0/OpenID Connect

To configure authentication with OAuth 2.0/OpenID Connect

1. In the Designer, select the **Base data > Security settings > OAuth 2.0/OpenID Connect configuration** category.
2. In the list editor, select the newly created identity provider.
3. Select the **General** tab and check the general configuration data of the identity provider.
 - **Column to search:** Select **ADSAccount - ObjectGUID**.
4. Select the **Applications** tab and check the configuration of the OAuth 2.0/OpenID Connect application.
 - **Default:** enabled
 - **Redirect URI:** If you want to use multifactor authentication with the administration tools of the One Identity Manager, enter **urn:InstalledApplication**.
5. Select the **Database > Commit to database** and click **Save**.

Related topics

- [Multi-factor authentication with One Identity Defender on page 138](#)
- [Configuring authentication with OAuth 2.0/OpenID Connect in the Web Portal on page 140](#)
- [Configuring RSTS for multi-factor authentication on page 139](#)
- [Creating the OAuth 2.0/OpenID Connect configuration on page 122](#)

Granular permissions for the SQL Server and database

For the deployment of a One Identity Manager database on an SQL Server, in Azure SQL Database or in a managed instance in Azure SQL Database, SQL Server logins and database users are provided for the administrative users, configuration users, and end users. Permissions at server and database level are matched to suit the user's tasks.

Normally, you cannot edit users and permissions.

For more information about users and their permissions, see the *One Identity Manager Installation Guide*, and the *One Identity Manager Data Archiving Administration Guide*.

Related topics

- [Displaying database users](#) on page 143
- [Displaying users' access levels](#) on page 144
- [Displaying database role and server role permissions](#) on page 144
- [Minimum access levels of One Identity Manager tools](#) on page 153

Displaying database users

To display information about a database user

1. In the Designer, select the **Base data > Security settings > Database server permissions > Database users** category.
2. Select the database user The following information is displayed:
 - **Login name:** The user's SQL Server login.
 - **Database server login:** Type of database user.
 - **Access level:** The access level for logging in. The access levels displayed are **End user**, **Configuration user**, **Administrative user**, **System administrator**, and **Unknown**.

3. To show assigned database roles and server roles, select the **Database or server role** tab.

Related topics

- [Displaying database role and server role permissions](#) on page 144

Displaying users' access levels

NOTE:

- If you select an existing database connection in the connections dialog, the access level of the login to be used is shown in a tooltip.
- Some user interfaces expect configuration user permissions at least. Logging in as an end user is not possible in this case.

To find the access level of the logged in user

- To display user information, double-click the icon in the program status bar 

On the **System user** tab, in the **SQL access level** field, you will see the access level for the current login. The access levels displayed are **End user**, **Configuration user**, **Administrative user**, **System administrator**, and **Unknown**.

Related topics

- [Displaying database users](#) on page 143

Displaying database role and server role permissions

Database role and sever role permissions are predefined and cannot be modified.

NOTE: The **End user role** database role is permitted for custom schema extensions.

To display database role and server role permissions

- In the Designer, select a database role or server role in the **Base data > Security settings > Database server permissions > Database and server roles** category.

This opens the List Editor showing each of the permissions.

Installing One Identity Redistributable Secure Token Server

The Redistributable Secure Token Server (RSTS) is a Secure Token Server component service designed to provide user authentication using standard federation protocols such as WS-Federation and OAuth 2.0. One Identity Manager uses the RSTS for authentication to web applications with Webauthn and OAuth 2.0.

For more information about the Webauthn configuration, see the *One Identity Manager Web Application Configuration Guide*.

Related topics

- [Installing One Identity Redistributable Secure Token Server](#) on page 145
- [Installing and uninstalling the One Identity Redistributable Secure Token Server](#) on page 147
- [Configuring RSTS for multi-factor authentication](#) on page 139

Installing One Identity Redistributable Secure Token Server

You can install multiple instances of the Redistributable Secure Token Server on one server.

To install the Redistributable Secure Token Server

1. Launch `autorun.exe` from the root directory of the One Identity Manager installation medium.
2. Switch to the **Other Products** tab
3. Select **One Identity Redistributable STS** and click **Install**.
4. On the start page of the installation wizard, click **Next**.

5. On the **Remove STS** page, you can uninstall an existing STS service or install another instance of the Redistributable STS.
| NOTE: This page is only shown if you have already installed an STS service.
 - Select **Install a new instance of Secure Token Server** to install another instance and click **Next**.
6. On the **Select database** page, select the One Identity Manager database connection. Select a user who has a minimum of administrative permissions for the database.
 - To use an existing connection to the One Identity Manager database, select it in the **Select a database connection** menu.
 - OR -
 - To create a new connection to the One Identity Manager database, click **Add new connection** and enter a new connection.
7. On the **Installation settings** page, enter the required information.
 - **Installation folder:** Select the directory where you want to install the STS.
 - **Signature certificate:** This certificate is used to sign authentication responses.
 - **SSL certificate:** This certificate is user for encrypting the SSL transport. If an SSL website is already setup, the SSL configuration of the IIS is used.
| NOTE: The **Redistributable STS Demo (for test purposes only)** certificate can be used only for test purposes.
 - **URL:** Specify the URL for reaching the STS.
 - **Client ID:** Enter the client ID.
 - **Configuration password and Confirm password:** Enter the password that you want to use later to access the STS configuration via the RSTS administration interface.
 - **User account:** The STS service runs under the user account given here. A domain account must be used to support Kerberos authentication. Ensure that the user account has the user permissions required to login as a service. Enter the user name and password.
8. On the **Identity provider** page, configure the RSTS to be the identity provider for One Identity Manager.
 - **Identity provider name:** Enter the name of the identity provider.
 - **Create a default OAuth 2.0/OpenID Connect application:** Specify whether to create and configure the identity provider in One Identity Manager. Enable this option to create an OAuth 2.0/OpenID Connect configuration.
9. On the **Installation** page you can see the installation progress. When the installation has finished, click **Next**.
10. Click **Finish** to close the installation wizard.

| NOTE: In a default installation, the service is entered in the server's services manager with the name **Redistributable Secure Token Server**. Further instances of the service

are entered in the server's services manager with a sequential number, for example **Redistributable Secure Token Server1, Redistributable Secure Token Server2.**

Related topics

- [Installing and uninstalling the One Identity Redistributable Secure Token Server on page 147](#)
- [OAuth 2.0/OpenID Connect on page 96](#)
- [OAuth 2.0/OpenID Connect authentication on page 120](#)
- [Multi-factor authentication with One Identity Defender on page 138](#)

Installing and uninstalling the One Identity Redistributable Secure Token Server

If you have multiple instances of a service installed, you can select the instance that you want to uninstall. To update the RSTS, uninstall the current service and the reinstall it again.

To uninstall the Redistributable Secure Token Server

1. Launch **autorun.exe** from the root directory of the One Identity Manager installation medium.
2. Switch to the **Other products** tab
3. Select **One Identity Redistributable STS** and click **Install**.
4. On the start page of the installation wizard, click **Next**.
5. On the **Remove STS** page, select the instance that you want to uninstall and click **Next**.
6. On the **Installation** page you can see the uninstall in progress. When the uninstall has finished, click **Next**.
7. Click **Finish** to close the installation wizard.

Related topics

- [Installing One Identity Redistributable Secure Token Server on page 145](#)

Preventing blind SQL injection

Due to security issues, you cannot run any database queries directly from the user interface or from web applications. Specific SQL operators undergo a risk assessment that prevents them from being used by One Identity Manager components. This includes operators such as LIKE, NOT LIKE, <, <=, >, or >=.

In order to continue using certain functions in One Identity Manager components, users require the **Common_AllowRiskyWhereClauses** program function.

Users who do not have this program function can only run database queries that are classified as trusted or pose no risk (risk index = **0.0**). Some of the functions in One Identity Manager components, such as testing dynamic roles or running filter queries, are not possible without this function.

If you want to allow certain users to run security-critical queries, you can assign permissions to users through permission groups.

- The **QBM_Critical_WhereClause** permissions group is provided for non role-based login. This group owns the program function. Add the system users who are allowed to run security-critical queries to the permissions group. Administrative system users automatically obtain these permissions groups.
- The **QER_4_Critical_WhereClause** permissions group is provided for non role-based login. This group owns the program function. The permissions group is linked to the **Base roles | security-critical queries** application role. Add the identities who are allowed to run security-critical queries in the application role.

Using configuration parameters, you can also control the risk assessment of running the SQL statements.

NOTE: The configuration parameters are effective only for users who have the **Common_AllowRiskyWhereClauses** program function.

- Use the **QBM | SQLCheck | RiskEvaluation** configuration parameter to define the risk assessment of running the SQL statements. Permitted values are:
 - **Low:** SQL statements with some risk are allowed.
 - **Medium:** The risk of SQL statements is assessed at a mitigated level. Thus, the threshold for blocking the user is reached later and more queries are possible.

- **Strict:** The risk of SQL statements is assessed in full. However, the user is not blocked until a certain threshold is reached.

If the configuration parameter is not set, the risk assessment is performed with the value **Strict**.

- Use the **QBM | SQLCheck | SubSelect** configuration parameter to specify how SQL statements with sub-queries are assessed. If the configuration parameter is set, then places where SQL statements with sub-queries are found are classified as higher risk.

Notes for customizations

- As an example, database queries that are required on customized forms or database queries that are run over the application server API, must be formulated as predefined database queries in One Identity Manager. Database queries are always run with the permissions of the current user. For more information about using predefined database queries, see the *One Identity Manager Configuration Guide*.
- You will find examples on the installation medium in the QBM\dvd\AddOn\ApiSamples directory.
- For the alphabetical display of objects such as identities or company structures, you can use the QERVFirstUnicodeChar table in customized menus.

Program functions for starting the One Identity Manager tools

The One Identity Manager tools can only be started if the user has the relevant program function permissions. The following program functions allow the One Identity Manager tools to be started.

To make the program function available to users

- In the Designer under the **Permissions > Program functions** category, check which permissions group contains the required program function and assign the program functions to other permissions groups as necessary.
- For non role-based login: Add the system user to the permissions group in the Designer under **Permissions > System users**.
- For role-based logins: Ensure that the user is assigned to the application role that owns the program function through its permissions group.

Table 41: Program functions for starting the One Identity Manager tools

Program function	Description
ApplicationStart_Analyzer	Allows the program Analyzer (Analyzer.exe) to be started.
ApplicationStart_ConfigWizard	Allows the program Configuration Wizard (ConfigWizard.exe) to be started.
ApplicationStart_CryptoConfig	Allows the program Crypto Configuration (CryptoConfig.exe) to be started.
ApplicationStart_DataImporter	Allows the program Data Import (DataImporter.exe) to be started.
ApplicationStart_DBClone	Allows the program (DBCclone.exe) to be started.
ApplicationStart_DBComparer	Allows the program (DBComparer.exe) to be started.
ApplicationStart_DBCompiler	Allows the program Database Compiler (DBCompiler.exe) to be started.
ApplicationStart_Designer	Allows the program Designer (Designer.exe) to be started.

Program function	Description
ApplicationStart_ JobQueueInfo	Allows the program Job Queue Info (JobQueueInfo.exe) to be started.
ApplicationStart_ LaunchPad	Allows the program Launchpad (LaunchPad.exe) to be started.
ApplicationStart_ LicenseMeter	Allows the program License Meter (LicenseMeter.exe) to be started.
ApplicationStart_Manager	Allows the program Manager (Manager.exe) to be started.
ApplicationStart_ ObjectBrowser	Allows the program Object Browser (ObjectBrowser.exe) to be started.
ApplicationStart_ OpSupport	Enables start-up of the Operations Support Web Portal.
ApplicationStart_ ReportEdit	Allows the program Report Editor (ReportEdit2.exe) to be started.
ApplicationStart_ SchemaExtension	Allows the program Schema Extension (SchemaExtension.exe) to be started.
ApplicationStart_ ServerInstaller	Allows the program Server Installer (ServerInstaller.exe) to be started.
ApplicationStart_ SoftwareLoader	Allows the program Software Loader (SoftwareLoader.exe) to be started.
ApplicationStart_ SynchronizationEditor	Allows the program Synchronization Editor (SynchronizationEditor.exe) to be started.
ApplicationStart_ SystemDebugger	Allows the program System Debugging (SystemDebugger.exe) to be started.
ApplicationStart_ Transporter	Allows the program Database Transporter (Transporter.exe) to be started.
ApplicationStart_ WebDesignerCompiler	Allows the program (VI.WebDesigner.CompilerCmd.exe) to be started.
ApplicationStart_ WebConfig	Allows the program Web Designer Configuration Editor (WebConfigEditor.exe) to be started.
ApplicationStart_ WebDesigner	Allows the program Web Designer (WebDesigner.exe) to be started.
ApplicationStart_ WebDesignerInstall	Allows the program Web Installer (WebDesigner.Installer.exe) to be started.

Related topics

- [Assigning program functions to permissions groups](#) on page 67
- [Adding system users to permission groups](#) on page 54

- [Assigning identities to application roles](#) on page [32](#)

Minimum access levels of One Identity Manager tools

NOTE:

- Connections that do not use the expected access level for SQL Server logins are not shown in the connection dialog.
- If you select an existing database connection in the connections dialog, the access level of the login to be used is shown in a tooltip.

You require the following minimum access level for One Identity Manager tools.

Table 42: Access level for One Identity Manager tools

Tool	Minimum access level
Analyzer	End user
Application server	End user or configuration user (depending on the application server's task)
API Server	End user
Configuration Wizard	Administrative user
Crypto Configuration	Configuration user
Data Import	End user Configuration user (saves import definition)
Database Transporter	Configuration user
Database Compiler	Configuration user
DBClone	Administrative user
DBComparer	Configuration user
Designer	Configuration user Some consistency checks require the administrative user access level.

Tool	Minimum access level
Job Queue Info	Configuration user
Launchpad	End user Some application that are started from the Launchpad, required different access levels
License Meter	End user
Manager	End user Some functions require configuration user access levels, for example, opening synchronization projects for target systems. Some consistency checks require the configuration user or administrative user access level.
Object Browser	End user
One Identity Manager Service	Configuration users for process collection with the MSSQLJobProvider
Report Editor	Configuration user
Schema Extension	Configuration user
Server Installer	Configuration user
Software Loader	Configuration user
Synchronization Editor	Configuration user
System Debugger	Configuration user
Web Designer	Configuration user
Web Designer Configuration Editor	Configuration user
Web Portal	End user
Password Reset Portal	End user
Operations Support Web Portal	End user
AppServer.Installer.CMD.exe	Configuration user
AutoUpdate.exe	Configuration user
DBCompilerCMD.exe	Configuration user
DBConsCheckCmd.exe	End user Some consistency checks require the configuration user or administrative user access level.

Tool	Minimum access level
DataImporterCMD.exe	End user
DBTransporterCMD.exe	Configuration user
Quantum.MigratorCmd.exe	Administrative user
SchemaExtensionCmd.exe	Configuration user
SoftwareLoaderCMD.exe	Configuration user
VI.WebDesigner.CompilerCmd.exe	Configuration user
WebDesigner.InstallerCMD.exe	Configuration user

Related topics

- [Granular permissions for the SQL Server and database](#) on page 143

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit <https://www.oneidentity.com/company/contact-us.aspx>.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at <https://support.oneidentity.com/>.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product

A

API Server

- OAuth 2.0/OpenID Connect 134

application role 9

- additional manager 19, 21-22

- administrators 11, 14, 16-19, 21-24, 26, 28

Application Governance 27

- administrators 27

- approver 27

- owner 27

- approver 19, 21

- approver (IT) 19, 21

- asset and account owners 27

- assign extended properties 36

- assign identity 32, 34

- assign reports 35

- attestation policy owner 17

- attestor for external users 17

- attestors 14, 16, 19, 21, 23

- auditors 14

- authorize as One Identity Manager administrator 29

- base roles 11, 13

- administrators 11, 29

- everyone (change) 11

- everyone (Change) 11

- identity managers 11

- internal permissions 11

- operational support 13

- password help desk 13

- synchronization post-

- processing 13

- certification status 37

- chief approval team 17, 23

- cloud administrators 26

- Compliance and Security Officer 13

- conflicting 35

- custom 28

- administrators 28

- manager 28

- dynamic 34

- edit 30-31

- exception approver 16

- extend permissions 33

- Identity and Access Governance 13-14, 16-18

- attestation 17

- administrators 17

- attestation policy owner 17

- chief approval team 17

- auditors 14

- company policies 16

- administrators 16

- attestors 16

- exception approver 16

- policy supervisors 16

- Compliance & Security Officers 13

- Identity Audit 14

- administrators 14

- attestors 14

- maintain SAP function 14

- rule supervisor 14
- subscribable reports 18
 - administrators 18
- Identity Management 19
 - application roles 22
 - additional manager 22
 - business roles 19
 - additional manager 19
 - administrators 19
 - approver 19
 - approver (IT) 19
 - attestors 19
 - identity
 - administrators 22
 - management level 19
 - organizations 21
 - additional manager 21
 - administrators 21
 - approver 21
 - approver (IT) 21
 - attestors 21
 - identity managers 11
 - internal permissions 11
 - management level 19
 - manager 31
 - overview 10
 - permissions group 31, 33
 - policy supervisors 16
 - Privileged Account Governance 27
 - product owners 23
 - put into operation 29
 - report 38
 - Request and Fulfillment 23
 - IT Shop 23
 - administrators 23
 - attestors 23
 - chief approval team 23
 - product owners 23
 - rule supervisor 14
 - self-registered identities 11
 - target system
 - administrators 24
 - target system managers 24
 - target system managers 24
 - Universal Cloud Interface
 - administrators 26
 - application server
 - OAuth 2.0/OpenID Connect 131
 - assignment resource
 - for an application role 37
 - authentication
 - test 118
 - authentication module
 - account-based system user 81
 - Active Directory user account 82
 - Active Directory user account (dynamic) 86
 - Active Directory user account (manual entry/role based) 85
 - Active Directory user account (manual) 84
 - Active Directory user account (role based) 83
 - assign application 107-108
 - component authenticator 99
 - Crawler 100
 - decentralized identity 104
 - decentralized identity (role-based) 105
 - enable 106
 - generic single sign-on (role based) 73

- HTTP header 93
- HTTP header (role based) 95
- identity 75
- identity (dynamic) 77
- identity (role-based) 76
- initial data 110
- LDAP user account (dynamic) 90
- LDAP user account (role-based) 87
- OAuth 2.0/OpenID Connect 96
- OAuth 2.0/OpenID Connect (role-based) 97
- password reset (role-based) 102
- reset password 100
- synchronization authenticator 98
- system user 73
- token 132
- user account 78
- user account (manual input/role-based) 80
- user account (role-based) 79
- Web Agent authenticator 99

C

- certification 37
- certification status 37

D

- database
 - authorizations 143
- database role
 - display permissions 144
- database server
 - access level 143
 - authorizations 143
 - database user 143

- login 143
- dynamic role
 - application role 34

E

- event
 - object event 69
 - program function 69

I

- identity
 - authorize as One Identity Manager administrator 29
- install RSTS 145

L

- Launchpad
 - action
 - program function 70

M

- method definition
 - program function 68
- Multi-factor authentication 137
 - Defender 138
 - OneLogin 137

O

- OAuth 2.0/OpenID Connect
 - API Server 134-135
 - application server 131-133
 - authentication 121
 - authentication module 96-97

- certificate 122
- configurations 120, 122, 128
- disabling columns 129
- enabling columns 129
- external application 133, 135
- identity provider 122, 128
- openid 122
- scope 122
- shared secret 122
- use case 122, 128
- Web application 127
- object
 - authorizations 63
- object event 69
 - program function 69

P

- permission
 - column permission 59
 - copy 60
 - database 143
 - database role 144
 - determine 43
 - edit 56
 - object 63
 - permissions filter 58
 - permissions group 57
 - rules 43
 - server role 144
 - simulation 62
 - SQL Server 143
 - table 57
 - table permissions 58
 - user 64
- Permissions Editor 56

- permissions group
 - assign application 64
 - authorizations 57
 - copy 48
 - dependencies 46-47
 - hierarchy 46
 - only for role-based login 50
 - predefined 40
 - program function 67-69
 - QBM_BaseRights 40
 - QER_OperationsSupport 40
 - role based 40
 - set up 45, 49-50
 - vi_4_ADMIN_LOOKUP 40
 - VI_4_ALLUSER 40
 - VI_Everyone 40
 - VI_View 40
 - vid 40
 - VID_Features 40
- program function 66-67, 69
 - Launchpad actions 70
 - method definition 68
 - permissions group 67-69
 - script 67

R

- RADIUS Server 138
- redistributable STS 145
- RSTS 138

S

- script
 - permission 67
 - program function 67

- Secure Token Server 145
- server role
 - display permissions 144
- system user
 - administrative user 53
 - determine dynamically 114
 - dynamic 40, 55
 - identity 55
 - logins 53
 - password 52-53
 - password never expires 52-53
 - permissions group 54
 - predefined 40
 - read-only permissions 53
 - sa 40
 - service account 53
 - set up 50-51
 - support 40
 - Synchronization 40
 - user 55
 - viadmin 40
 - viHelpdesk 40
 - configuration data 114
- user
 - access level 144
 - authentication module 64
 - authorizations 64
 - dynamic 64
 - permissions group 64
 - program function 64
 - read permissions 64
 - system user 64

T

- table
 - authorizations 57
- token
 - authentication module 132

U

- use case
 - assign authentication module 107-108
 - assign permissions group 64