One Identity Manager 8.1.3

Authorization and Authentication Guide

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

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

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

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

Patents

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

Trademarks

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

Legend

> ⊗ **WARNING: A WARNING icon 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 - June 2020
Version - 8.1.3

Contents

About this guide

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

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.

You will find an overview of the default application roles, default permissions groups and system users of One Identity Manager. You will learn how to get the application roles up and running. 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.

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.

2

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, employees, company resources, approval workflows and approval policies are assigned to fixed application roles. Edit permissions for these application roles do not need to be defined specifically for the company. This simplifies administration of edit permissions.
- Enables audit secure internal administration of One Identity Manager users and their write permissions. Edit permissions can be granted through assignment, request, and approval or by calculation on account of specific properties. The plausibility of the edit 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 limited to permissions groups whose edit permissions are predefined by One Identity Manager. Controlling edit permissions:

- Navigation configuration in administration tools
- Access to objects and their properties
- 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 9
- Implementing the application roles on page 24
- Creating and editing application roles on page 25

Related topics

- Granting One Identity Manager schema permissions through permissions groups on page 33
- One Identity Manager authentication modules on page 67

Application roles overview

One Identity Manager supplies default application roles whose permissions are matched to the different task and functions. Assign employees 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.

The following default application roles are defined:

- Application roles for basic functions
- Compliance and security officer
- Auditors
- Application roles for identity audit
- Application roles for company policies
- Application roles for attestation
- Application roles for subscribable reports
- Management level
- Application roles for business roles
- Application roles for organizations
- Application roles for employees
- Application roles for the IT Shop
- Application roles for target systems
- Application roles for Universal Cloud Interface
- Application roles for custom tasks

Application roles for basic functions

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	Administrators must be assigned to the **Base roles | Administrators** application role. Users with this application role: - Administer application roles for administrators. - Assign employees to administrator application roles. - Add other employees to the **Base roles | Administrators** application role and edit conflicting application roles. - See the master data for the other application roles. - Can use Password Reset Portal to set passwords for selected system users.
Everyone (change)	The **Base roles | Everyone (change)** application role is automatically assigned to every user. Users with this application role: - Can edit certain employee master data in the Web Portal. Should every user be automatically assigned to a custom permissions group when they log in, then this permissions group can be added to the application role. Members of this application role are determined through a dynamic role.
Everyone (lookup)	The **Base roles | Everyone (Lookup)** application role is automatically assigned to every user. Users with this application role: - Obtain read access to objects in the Web Portal. Should every user be automatically assigned to a custom permissions group when they log in, then this permissions group can be added to the application role. Members of this application role are determined through a dynamic role.
Employee managers	The **Base roles | Employee managers** application role is automatically assigned to a user if the user is a manager or supervisor of employees, departments, locations, cost centers,

Application role	Description
	business roles, or IT Shops. Users with this application role: • Can edit master data for the objects they are responsible for and assign company resources to them. • Can edit new employees added in the Web Portal and edit the master data of their staff. • Can add their staff members to the IT Shop. • Can view their staff's compliance rule violations in the Web Portal. Members of this application role are determined through a dynamic role.
Birthright Assignments	The **Base roles | Birthright assignments** application role is used to provide birthrights to employees which are provided to establish their working environment. The application roles are allocated all the resources marked for automatic assignment to all employees. All internal employees are assigned to this application role and obtain the resources. Internal employees are found through a dynamic role.
Operations support.	Employees that use the Operations Support Web Portal, must be assigned the **Base roles** | **Operations support** application role. Users with this application role: • Monitor handling of Job queue processes • Monitor handling of the DBQueue • Create access codes to enable employees to log on to Password Reset Portal
Self-registered employees	All new external employees that have registered themselves in the Web Portal, are assigned to the **Base roles | Self-registered employees** application role. These employees are determined by a dynamic role.

Related topics

- Customized extension of application role edit permissions on page 28

Compliance and security officer

> 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 polices.

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 2: Application roles for identity audit

Application role	Description
Administrators	Administrators must be assigned to the **Identity & Access Governance | Identity Audit | Administrators** application role. Users with this application role: • 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
	- Set up other application roles as required.
Rule supervisors	Rule supervisors must be assigned to the **Identity & Access Governance | Identity Audit | Rule supervisors** application role or a child application role. Users with this application role: - 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	Administrators must be assigned to the **Identity & Access Governance | Identity Audit | Exception approvers** application role or a child application role. Users with this application role: - Edit rule violations in the Web Portal. - Can grant exception approval or revoke it in the Web Portal.
Attestors	Attestors must be assigned to the **Identity & Access Governance | Identity Audit | Attestors** application role. Users with this application role: - Attest compliance rules and exception approvals in the Web Portal for which they are responsible. - Can view master data for these compliance rules but not edit them. NOTE: This application role is available if the module Attestation Module is installed.
Maintain SAP Functions	Administrators must be assigned to the **Identity & Access Governance | Identity Audit | Maintain SAP functions** application role or a child application role. Users with this application role: - 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
	- Assign mitigating controls. NOTE: This application role is available if the module SAP R/3 Compliance Add-on Module is installed.

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 3: Application roles for company policies

Application role	Description
Administrators	Administrators must be assigned to the **Identity & Access Governance | Company policies | Administrators** application role. Users with this application role: - 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	Policy supervisors must be assigned to the **Identity & Access Governance | Company policies | Policy supervisors** application role or another child application role. Users with this application role: - 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.
Exception	Exception approvers must be assigned to the **Identity & Access**

Application role	Description
approvers	**Governance | Company policies | Exception approvers** application role or a child application role. Users with this application role: - Edit policy violations. - Can grant exception approval or revoke it.
Attestors	Attestors must be assigned to the **Identity & Access Governance | Company policies | Attestors** application role. Users with this application role: - Attest company policies and exception approvals in the Web Portal for which they are responsible. - Can view the master data for these company policies but not edit them. NOTE: This application role is available if the module Attestation Module is installed.

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 4: Application roles for attestation

Application role	Description
Administrators	Administrators are assigned to the **Identity & Access Governance | Attestation | Administrators** application role. Users with this application role: - 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. - Monitor attestation cases.

Application role	Description
Chief approval team	The chief approver must be assigned to the **Identity & Access Governance | Attestation | Chief approval team** application role. Users with this application role: • Approve using attestation cases. • Assign attestation cases to other attestors.
Attestors for external users	Attestors for external users must be assigned to the **Identity & Access Governance | Attestation | Attestors for external users** application role. Users with this application role: • Attests new, external employees.

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 5: Application roles for subscribable reports

Application role	Description
Administrators	Administrators must be assigned to the **Identity & Access Governance | Company policies | Report Subscriptions** application role. Users with this application role: • Create subscribable reports from existing reports. • Configure report parameters for subscribable reports. • Assign subscribable reports to employees, company structures or IT Shop shelves. • Create custom mail templates for sending subscribed reports by email.

Management level

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

Users with this application role:

- Can view reports and statistics for management levels in the Web Portal.

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 6: Application roles for business roles

Application role	Description
Administrators	Administrators must be assigned to the **Identity Management | Business roles | Administrators** application role. Users with this application role: • Create and edit business roles. • Assign company resources to business roles. • Administrate application roles for role approvers, role approvers (IT), and attestors. • Set up other application roles as required.
Attestors	Attestors must be assigned to the **Identity Management | Business roles | Attestors** application role or a child application role. Users with this application role: • Attest correct assignment of company resource to business roles for which they are responsible. • Can view master data for these business roles but not edit them. NOTE: This application role is available if the module Attestation Module is installed.
Role approver	Approvers must be assigned to the **Identity Management | Business roles | Role approvers** application role or a child application role. Users with this application role: • Are approvers for the IT Shop. • Approve requests from business roles for which they are respons-

Application role	Description
	ible.
Role approver (IT)	IT role approvers must be assigned to the **Identity Management | Business roles | Role approvers (IT)** application role or a child application role. Users with this application role: • Are IT role approvers for the IT Shop. • Approve requests from business roles for which they are responsible.

Application roles for organizations

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

Table 7: Application roles for organizations

Application role	Description
Administrators	Administrators must be assigned to the **Identity Management | Organizations | Administrators** application role. Users with this application role: • Set up and edit departments, cost centers, and locations. • Assign company resources to departments, cost centers, and locations. • Administrate application roles for role approvers, role approvers (IT), and attestors. • Set up other application roles as required.
Attestors	Attestors must be assigned to the **Identity Management | Organizations | Attestors** application role or a child application role. Users with this application role: • Attest correct assignment of company resources to departments, cost centers, and locations for which they are responsible. • Can view master data for departments, cost centers, and locations but cannot edit them. NOTE: This application role is available if the module Attestation Module is installed.

Application role	Description
Role approver	Role approvers must be assigned to the **Identity Management | Organizations | Role approvers** application role or a child application role. Users with this application role: • Are approvers for the IT Shop. • Approve request from departments, cost centers, and locations for which they are responsible.
Role approver (IT)	IT role approvers must be assigned to the **Identity Management | Organizations | Role approvers (IT)** application role or a child application role. Users with this application role: • Are IT role approvers for the IT Shop. • Approve request from departments, cost centers, and locations for which they are responsible.

Application roles for employees

The following application role is available for employee administration:

Table 8: Application roles for employees

Application role	Description
Administrators	Employee administrators must be assigned to the **Identity Management | Employees| Administrators** application role. Users with this application role: • Can edit master data for all employees • Can assign a manager. • Can assign company resources to employees. • Check and authorize employee master data. • Create and edit risk index functions. • Edit password policies for employee passwords • Delete employee's security keys (WebAuthn)

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 9: Application roles for the IT Shop

Application role	Description
Administrators	Administrators must be assigned to the **Request & Fulfillment | IT Shop | Administrators** application role. Users with this application role: • 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. • 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 and employees. • Assign system entitlements to IT Shop structures.
Product owners	Product owners must be assigned to the **Request & Fulfillment | IT Shop | Product owners** application role or a child application role. Users with this application role: • Approve through requests. • Edit service items and service categories under their management.
Attestors	Attestors must be assigned to the **Request & Fulfillment | IT Shop | Attestors** application role. Users with this application role: • Attest correct assignment of company resource to IT Shop structures for which they are responsible.

Application role	Description
	- Can view master data for these IT Shop structures but not edit them. NOTE: This application role is available if the module Attestation Module is installed.
Chief approval team	Chief approvers must be assigned to the **Request & Fulfillment | IT Shop | Chief approval team** application role. Users with this application role: - Approve through requests. - Assign requests to other approvers.

NOTE: Approvers in charge 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 10: Application roles for target systems

User	Tasks
Target system administrators	Target system administrators must be assigned to the **Target systems | Administrators** application role. Users with this application role: - 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 employees to be target system administrators. - Do not assume any administrative tasks within the target

User	Tasks
	system.
Target system managers	Target system managers must be assigned to the **Target systems | <target system>** application role or a child application role. 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. Users with this application role: • Assume administrative tasks for the target system. • Create, change, or delete target system objects like user accounts or groups. • Edit password policies for the target system. • Prepare system entitlements to add to the IT Shop. • Can add employees who have an other identity than the **Primary identity**. • 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 employees within their area of responsibility as target system managers and create child application roles if required.
Target system managers for Unified Namespace	Target system managers must be assigned to the **Target systems | Unified Namespace** application role or a child application role. Users with this application role: • Obtain view of the objects in the connected target systems across all target systems. • Can create reports across all target systems. If the users are also target system managers of the basic underlying target systems, you can manage these target systems through the Unified Namespace.

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 11: Application roles for Universal Cloud Interface

User	Tasks
Cloud administrators	Administrators must be assigned to the **Universal Cloud Interface | Administrators** application role or a child application role. Users with this application role: • 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.
Cloud operators	Operators must be assigned to the **Universal Cloud Interface | Operators** application role or a child application role. Users with this application role: • Edit pending, manual provisioning processes in the Web Portal and obtain statistics.
Cloud auditors	Auditors must be assigned to the **Universal Cloud Interface | Auditors** application role or a child application role. Users with this application role: • Can view manual provisioning processes in the Web Portal and obtain statistics.

Application roles for custom tasks

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

Table 12: Application roles for custom tasks

Application role	Description
Administrators	Administrators must be assigned to the **Custom | Administrators**

Application role	Description
	application role. Users with this application role: - Administrate custom application roles. - Set up other application roles for managers if required.
Manager/supervisor	Managers must be assigned to the **Custom | Managers** application role or a child role. Users with this application role: - Add custom task in One Identity Manager. - Configure and start synchronization in the Synchronization Editor. - Edit the synchronization's target system types as well as outstanding objects in the Manager. You can use these application roles, for example, to guarantee One Identity Manager users write permissions on custom tables or columns. All application roles that you define here must obtain their write permissions through custom permissions groups.

Implementing the application roles

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

Run this task once.

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

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

NOTE: Once you update the view in 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 employees assigned to this application role.

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

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

Related topics

- Assigning employees to application roles on page 27
- Creating and editing application roles on page 25

Creating and editing application roles

To set up your first application roles you need to add an employee to the application role **Base roles | Administrators**. This employee is authorized to add more employees to different administration application roles. For more information, see Implementing the application roles on page 24.

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

> 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 master data** task.
3. Edit the application role's master 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 master data.
4. Save the changes.

> NOTE: You cannot delete default application roles.

Related topics

- Master data of application roles on page 26
- Assigning employees to application roles on page 27
- Customized extension of application role edit permissions on page 28
- One Identity Manager authentication modules on page 67

Master data of application roles

Table 13: 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.
Permissions group	Permissions group for determining write permissions on role-based login. The application role is given access permissions of the associated permissions group. If there is no permissions group assigned, the application role gets write permissions from the parent application role. Administrators can assign the rest of the application roles to custom defined permissions groups. For more information, see Customized extension of application role edit permissions on page 28. NOTE: Permissions groups for default administrator application roles for cannot be edited.
Description	Text field for additional explanation.
Comment	Text field for additional explanation.
Certification status	Status of the application role's certification. The following values can be selected. • **New**: The application role was newly created in the One Identity Manager database. • **Certified**: The master data of the application role is approved by a manager. • **Denied**: The application role master data was not approved by a manager.

Property	Meaning
Block inheritance	Specifies whether inheritance for this application role can be discontinued. Set this option to prevent company resources being inherited by child application roles.
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.

Assigning employees to application roles

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

Employees of the parent application role are inherited if no employees are directly assigned to an application role.

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

To assign employees to an application role

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

   > TIP: In the **Remove assignments** pane, you can remove assigned employees.
   >
   > ***To remove an assignment***
   >
   > - Select the employee and double-click ✓.

4. Save the changes.

Related topics

- Creating and editing dynamic roles for application roles on page 29

Customized extension of application role edit permissions

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

Some of the default application roles are already assigned permissions groups. These permissions groups have the edit 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 write 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. Create a new permissions group in the Designer.

   > 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. As a result, 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 edit permissions.

Related topics

- Master data of application roles on page 26
- Granting One Identity Manager schema permissions through permissions groups on page 33

Additional tasks for managing application roles

After you have entered the master data, you can run the following tasks.

Creating and editing dynamic roles for application roles

Use this task to assign employees to an application role through dynamic roles. For detailed 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, which 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 master data. The following applies to dynamic roles for application roles:
   - **Object class**: Select **Employee**.
   - **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 master data** task.
5. Edit the dynamic role.
6. Save the changes.

Related topics

- Master data of application roles on page 26

Specifying mutually exclusive application roles

It is possible that employees 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 person anymore.

> NOTE: Only system roles, which are defined directly as conflicting application roles cannot be assigned to the same employee. 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 the application roles that are mutually exclusive to the selected application 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 employee in this application role can subscribe to reports in the Web Portal. For detailed information about subscribable reports, 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 | Employee Managers**, 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.

   - OR -

   In the **Remove assignments** pane, remove the reports.
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 detailed 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 detailed 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, which takes you through adding an assignment resource.

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 14: 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 employees from the selected application role are also members. For detailed information about analyzing role memberships, see the *One Identity Manager Identity Management Base Module Administration Guide*.
Show historical memberships	This report lists all members of the selected application role and the length of their membership.

3

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 edit permissions to the 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 34
- Rules for determining the valid permissions for tables and columns on page 36
- Processing permissions groups on page 39
- Editing system users on page 44
- Editing table permissions and column permissions on page 49
- Managing permissions to program features on page 60
- Displaying permissions for objects on page 57
- Displaying permissions for the current user on page 58
- Assigning permissions groups to applications on page 58

Related topics

- One Identity Manager application roles on page 8
- One Identity Manager authentication modules on page 67

Predefined permissions groups and system users

One Identity Manager provides permissions groups and system users with a predefined user interface and special edit 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 15: 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 executing special functions in One Identity Manager.
Permissions group **VI_View**	The **VI_View** permissions group has viewing permissions for all tables and columns of the One Identity Manager application data model. NOTE: Assign viewing permissions of custom schema extensions to the permissions group.
Permissions group **VI_Everyone**	The **VI_Everyone** permissions group is assigned to form 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 the One Identity Manager application data model	Permissions groups have edit permissions for One Identity Manager application data model tables and columns. These permissions groups are equipped with menu items, forms, tasks, and program functions which allows the application data to be edited with the Manager.

Permissions group	Description
Permissions groups for the One Identity Manager system data model	These permissions groups have permissions for the One Identity Manager system data model tables and columns. 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. The **vid** permissions group has all edit permissions for the system configuration with the Designer.
Role-based permissions group **VI_4_ALLUSER**	The **VI_4_ALLUSER** permissions group provides the base rights as well as menu items, forms, task, 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**	The **vi_4_ADMIN_LOOKUP** permissions group has the viewing permissions for all tables and columns of the One Identity Manager application data model. NOTE: Assign viewing permissions of custom schema extensions to the permissions group. Assign viewing permissions of the module's own tables and columns to the permissions group.
Role-based permissions group **QER_OperationsSupport**	The **QER_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 edit permissions for One Identity Manager application data model tables and columns. 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 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 16: Predefined system users

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

System users	Description
System user **sa**	The **sa** system user is used exclusively by the One Identity Manager Service. This system user is not allocated a permissions groups but has all access permissions, tasks, and program functionality.
System user **viadmin**	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. IMPORTANT: Do not use the **viadmin** system user in a live environment. Create your own system user with the appropriate permissions. 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 employee as a One Identity Manager administrator for the role-based login. The system user is not a member of the application role themselves.
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 helpdesk resources.

Related topics

- Deleting dynamic system users on page 49
- Permissions group dependencies on page 40
- Processing permissions groups on page 39

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 are 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 database system files are granted by the system, which are sufficient for logging a system user into One Identity Manager 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 references table, the primary key column and the columns that are necessary on 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 of permissions grouping 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 of limiting conditions

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

Permissions group	Viewing Condition	Column Viewing Permissions
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 employee objects.

Person.Lastname	Visible Columns
Smith	Lastname
Bishop	Lastname, Firstname, Entrydate
Bennett	Lastname, Firstname, Gender
Dummy	Lastname

Processing permissions groups

One Identity Manager provides permissions groups with a predefined user interface and special edit permissions for the 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 summarized 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 in User & Permissions Group Editor. 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:

- Editing the master data of a permissions group
- Defining new permissions group dependencies
- Copying Permissions Groups
- Creating a new permissions group

Related topics

- Predefined permissions groups and system users on page 34
- Permissions groups properties on page 40
- Editing the dependencies of permissions groups on page 41

- Copying permissions groups on page 42
- Creating permissions groups on page 44
- Managing permissions to program features on page 60

Permissions groups properties

Table 17: Permissions group properties

Property	Description
Permissions group	Name of the permissions group. Label custom permission groups with the **CCC** prefix.
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 42
- Creating permissions groups on page 44

Permissions group dependencies

You can enable permissions and user interface components to be passed on from one permissions group to other permissions groups by structuring permissions groups hierarchically. 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 permissions groups are defined with the following permissions and user interface components.

Permissions group	Permissions	User interface
A	Viewing permissions	Menu structures and forms
B	Edit permissions	Task definitions

Permissions group B is arranged 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

- Editing the dependencies of permissions groups on page 41

Editing the dependencies of permissions groups

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, the dependencies to other permissions groups are marked in color which also is also used to show the direction of inheritance.

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

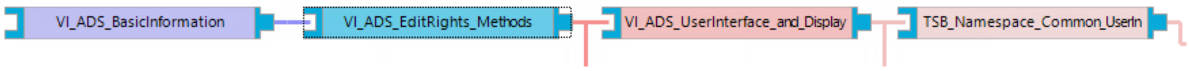
VI_ADS_BasicInformation
VI_ADS_EditRights_Methods
VI_ADS_UserInterface_and_Display
TSB_Namespace_Common_UserIn

Table 18: 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.
Red	This permissions group is a parent of the selected permissions group and passes inheritance to 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 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. Permissions subgroups inherit permissions from the selected permissions group.

Copying permissions groups

The User & Permissions Group Editor provides a wizard for copying edit 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.
4. On the home 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 19: 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: Note here that 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 44
- Permissions groups properties on page 40
- Editing the dependencies of permissions groups on page 41
- Adding system users to permission groups on page 47
- Editing table permissions and column permissions on page 49

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 master data for the permissions group.
5. Save the changes.

Related topics

- Copying permissions groups on page 42
- Permissions groups properties on page 40
- Editing the dependencies of permissions groups on page 41
- Adding system users to permission groups on page 47

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 data model, 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 permissions of the **viadmin** default system user.

In the Designer, system users are shown in the **Permissions | System users** category. You will see 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 a new system user, such as an administrative system user or a system user for service accounts
- Configuration of password settings for system users

- Adding a system user to permissions groups
- Determining which employees use a system user

> NOTE: You cannot edit dynamic system users.

Related topics

- Predefined permissions groups and system users on page 34
- Creating system users on page 45
- System users’ passwords on page 46
- System user properties on page 46
- Adding system users to permission groups on page 47
- Which employees use the system user? on page 48
- Deleting dynamic system users on page 49

Creating system users

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 master data.
5. Add the system user to permissions groups.
6. Save the changes.

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

Related topics

- Predefined permissions groups and system users on page 34
- System users’ passwords on page 46
- System user properties on page 46
- Adding system users to permission groups on page 47
- Deleting dynamic system users on page 49
- Which employees use the system user? on page 48
- One Identity Manager authentication modules on page 67

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 access code for a one off log in on the Web Portal (`Person.Passcode`).

If necessary, adjust the password policy to your requirements in the Designer. For detailed information about editing password policies, see *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 employees, 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 46

System user properties

Table 20: 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 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 of all permissions groups, but should only have viewing permissions for the objects. This results in overwriting all other edit permissions that the system user obtains through permissions group memberships.
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

Property	Description
	required if the **Account based system user** authentication module is used to log into the One Identity Manager tools.
Administrative user	Specifies whether this is an administrative system user. Administrative system users are automatically added to all non role-based permissions groups. NOTE: You can create an administrative system user in the User & Permissions Group Editor using the **Create administrator** menu.
Service account	Specifies whether this is a system user that is used by a service account. This system user is not allocated a permissions groups but has all access 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.

Related topics

- System users' passwords on page 46

Adding system users to permission groups

Add the system user to permissions groups, thereby granting the system user permissions for the tables and columns of the One Identity Manager data model 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 permissions 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 of permissions groups are presented 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 system users.

Figure 2: System user permissions group memberships

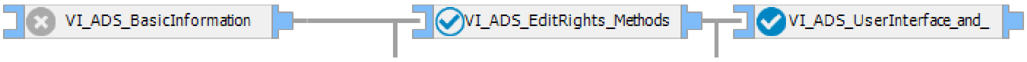

Table 21: Meaning of icons in the hierarchical display

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. Select the required permissions group in the hierarchical view. By clicking on the icon you add or delete the selected system user to or from a permissions group.

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

Related topics

- Deleting dynamic system users on page 49

Which employees use the system user?

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

To display which employees 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 employees** menu item.

   > NOTE: You cannot change the assignments in this view.

Deleting dynamic system users

> NOTE: If no role-based logins of employees who 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 employee 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.

Editing table permissions and column permissions

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:

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

The permissions for predefined permission groups to predefined tables and columns in the One Identity Manager schema cannot be changed.

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

Detailed information about this topic

- Rules for determining the valid permissions for tables and columns on page 36
- Displaying the permissions of a permissions group on page 50
- Displaying permissions for tables on page 50
- Editing table properties on page 51
- Editing column permissions on page 53

- Copying table permissions and column permissions on page 54
- Simulating permissions for system users on page 55

Displaying the 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 for which you want to display permissions.

   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 detailed information about creating user-defined filters in the Designer, see the *One Identity Manager User Guide for One Identity Manager Tools User Interface*.

Displaying permissions for tables

In the **Summary of all permissions** view in the Permissions Editor, the permissions groups that have permissions for a table or column are displayed. 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**. 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 permissions for a column, open the table entry in the upper area of 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 properties

Use the table permissions to grant the 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 employees to their last names. For instance, a user can be given read access only to the employees whose last names begin with A-F, whereas he/she can edit employees with last names beginning with G-Z.

> NOTE: The 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 permissions for a table 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.

   > TIP: Use **Shift + select** or **Ctrl + select** to select multiple tables.

5. Edit the permissions for the permissions group in the **Permissions** area.
   - To insert new permissions, select the **New** context menu and enable the associated check boxes. You can grant the following permissions.

Table 22: Table permissions

Permissions	Meaning
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 a permission, disable the associated checkbox.
- Use the **Delete** context menu, to withdraw all permissions for a table.

6. (Optional) To specify other conditions for table permissions, go to the lower part of the Permissions Editor and change to the **Group permissions for table** view and select the **Permissions filter** tab.

   > NOTE: You can only define permissions filters for the tables of the application data model.

   - Enter the conditions as valid WHERE clauses for database queries. You can enter the following permissions filters.

Table 23: Permissions filter

Permissions filter	Meaning
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 for permissions filters

A user should be able to see all employees, but only edit the employees 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.

Related topics

- Editing column permissions on page 53
- Copying table permissions and column permissions on page 54

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** permissions is required for at least the required fields in the table.
> - NOTE: If you grant the **Insert** or **Edit** permissions, the **View** permission is also granted.
> - Use the column definition to conditionally remove viewing permissions from scripts or create edit permissions for a column. If the return value is **False**, the permissions are removed. For more information about editing column definitions, see the *One Identity Manager Configuration Guide*.

> NOTE: The 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 modify the permissions for a column 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. Edit the permissions for the permissions group in the **Permissions** area.
   - To insert new permissions, select the **New** context menu and enable the associated check boxes. You can grant the following permissions.

     Table 24: Column permissions

Permissions	Meaning
Viewable	The column is displayed.

Permissions	Meaning
Editable	The values in the columns can be changed
Insertable	The value in the column can be edited when a new data record. Once the data record has been saved it can no longer be edited. For example, when an Active Directory User is created, an Active Directory container is defined. Because this is a key field the Active Directory container cannot be changed after the object has been saved.

- To withdraw a permission, disable the associated checkbox.
- To withdraw all permissions for a column, use the **Delete** context menu.

Related topics

- Editing table properties on page 51
- Copying table permissions and column permissions on page 54

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 methods copies the permissions of the source table (source column) to 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). Any existing rights for the target table (target column) remain unaffected.
- **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 the permissions of 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. 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 to copy the permissions to the buffer.
   c. Select the table at the top of the Permissions Editor for 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 copy permissions.
   d. Use the **Insert** context menu to insert the permissions.
   e. If necessary, repeat step c) and d) for other columns.

Related topics

- Editing table properties on page 51
- Editing column permissions on page 53

Simulating permissions for system users

By simulating the permissions in the Permissions Editor, you can see which permissions a system user has based on his or her 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 permissions simulation

1. In the Designer, select the **Permissions** category.
2. Start the Permissions Editor using the **Edit permissions** 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 tables.

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**
    >
    > For the `Employee` table, viewing permissions to the `UID_Org` column have been assigned. As a result, viewing permissions are implicitly assigned for columns of the `Org` table, which are used as a display template, for example, `Org.Ident_Org`.
    >
    > To simulate this example, select the **Employee 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 area of the Permissions Editor in the **Simulation** area.

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 area of the Permissions Editor.

    The permissions and permissions groups are displayed in the **Permissions simulation** view in the lower area of 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

You can display object properties and permissions in One Identity Manager tools.

To show extended object properties

- Select the object and open the **Properties** context menu.

On **General**, you can see the object‘s general properties, for example, ID, status, or primary key.

All the object columns are displayed in a grid on **Properties** with their values. You can choose between a simple column view and the advanced view with additional data for column definitions.

Table 25: Icon used for column properties

Icon	Meaning
▸	Required field.
🟥	No viewing permissions.
■ ▸	No edit permissions.

On **Access permissions**, you can see which permissions are valid for an object based on permissions groups. 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 additional user information, double-click the 👤 icon in the 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.
Employee UID (UserUID)	Unique ID for the current user’s employee if an employee related 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.
Remarks	More details about the system user in use.
Permissions groups	Permissions groups that are assigned to the system user. Which user interface and editing permissions apply depend on the permissions groups.
Program functions	Program functions assigned to the system user The menu items and functions available depend on the program functions.

Assigning permissions groups to 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 permissions group to an application

1. In the Designer, select the **Permissions | Permissions groups | Role based permissions groups** category.
2. Select the **View | Select table relations** menu item and enable the `DialogGroupInProductLimited` table.
3. In the List Editor, select the permissions group.
4. Assign the application in the **Applications** edit view.

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

4

Managing permissions to program features

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

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

- Program functions for starting the One Identity Manager tools on page 60
- Displaying permissions for the current user on page 58
- Assigning program function to permissions groups on page 63
- Permissions for executing scripts on page 63
- Permissions for executing methods on page 64
- Permissions for triggering processes on page 65
- Modifying permissions for executing actions in the Launchpad on page 66

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

Table 28: Program functions for starting the One Identity Manager tools

Program function	Description
ApplicationStart_Analyzer	Allows the program Analyzer (`Analyzer.exe`) to be started.
ApplicationStart_ApiDesigner	Allows the program API Designer (`ApiDesigner.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 (`DBClone.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.
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_	Allows the program Report Editor (`ReportEdit2.exe`) to be

Program function	Description
ReportEdit	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 function to permissions groups on page 63
- Adding system users to permission groups on page 47
- Assigning employees to application roles on page 27

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 function to permissions groups

To assign a program function to a permissions group

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. In the **Permissions groups** edit view, assign the Permissions groups.
5. Save the changes.

Related topics

- Adding system users to permission groups on page 47

Permissions for executing scripts

The basic permissions for executing scripts are granted to the logged in user through the program feature **Allow the starting of arbitrary scripts from the frontend** (`Common_StartScripts`).

If a script is assigned a program function (`QBMScriptHasFeature` table), users can only execute 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 execution of a script 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 trigger 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+Selection** to select your newly created program function and the **Allow the starting of arbitrary scripts from the frontend** function (`Common_StartScripts`).
   e. Assign the permissions group in the **Permissions groups** edit view.

Related topics

- Processing permissions groups on page 39

Permissions for executing methods

If a task definition is assigned a program function (`QBMMethodHasFeature` table) users can only execute 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 required program functions to the custom permissions group whose systems users will trigger these tasks.
    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 **Permissions groups** edit view, assign the permissions group.

Related topics

- Processing permissions groups on page 39

Permissions for triggering processes

The basic permissions for triggering processes are granted to the logged in user by the **Allow to trigger any events from the frontend** program feature (Common_TriggerEvents).

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). If the object event is assigned a program function (QBMEventHasFeature table), users that own this program function by permissions group, can trigger the object event and therefore the process, irrespective of their permissions.

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. Assign the object events in the **Object events** edit view.

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 List Editor, use **Ctrl+Selection** to select your newly created program function and the function **Allow to trigger any events from the frontend** (Common_TriggerEvents).

   d. In the **Permissions groups** edit view, assign the permissions group.

Related topics

- Processing permissions groups on page 39

Modifying permissions for executing 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. Save the changes.

5

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 employee memberships in the One Identity Manager application roles are determined during login. Assignments of permissions group to One Identity Manager application roles are used to determine which permissions groups apply to the employee. A dynamic system user is determined from these permissions groups that will be used for the employee’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.

System users

Credentials	The system user's identifier and password.
Prerequisites	• 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	The user interface and the write permissions are loaded through the system user. Data modifications are attributed to the system user.

IMPORTANT: The **viadmin** system user is available by default. The system user has the predefined user interface and access permissions to database resources. The user interface and the permissions structure for the system user must not be used or changed in the production system because this system user is overwritten as a template user with each schema update.

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	• The employee exists in the One Identity Manager database. • The employee is assigned at least one application role. • The user account exists in the One Identity Manager database and the employee is entered in the user account's master data.
Set as default	No

Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	One Identity Manager searches for the user account according to the configuration and finds the employee assigned to the user account. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. A dynamic system user determined from the employee's application roles. The user interface and the write permissions are loaded through this system user. Changes to the data are assigned to the logged in employee.

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

Table 29: Configuration parameters for the authentication module

Configuration parameter	Meaning
QER | Person | GenericAuthenticator	This configuration parameter specifies whether authentication through single sign-on is supported.
QER | Person | GenericAuthenticator | SearchTable	This configuration parameter contains the table in the One Identity Manager schema in which user information is stored. The table must contain a foreign key with the name `UID_Person`, which points to the table `Person`. Example: `ADSAccount`
QER | Person | GenericAuthenticator | SearchColumn	This configuration parameter contains the column from the One Identity Manager table (SearchTable), which is used to search for user name of the current user. Example: `CN`
QER | Person | GenericAuthenticator | EnabledBy	This configuration parameter contains a pipe (|) delimited list of Boolean columns from the One Identity Manager table (SearchTable), which enables the user account for login.

Configuration parameter	Meaning
QER | Person | GenericAuthenticator | DisabledBy	This configuration parameter contains a pipe (|) delimited list of Boolean columns from the One Identity Manager table (SearchTable), which disables the user account for login. Example: `AccountDisabled`

Employee

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

Credentials	Employee's central user account and password.
Prerequisites	• The system user with permissions exists in the One Identity Manager database. • The employee exists in the One Identity Manager database. • The central user account is entered in the employee's master data. • The system user is entered in the employee's master data. • The system user password is entered in the employee's master data.
Set as default	Yes
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. The user interface and the write permissions are loaded through the system user that is directly assigned to the logged in employee. Changes to the data are assigned to the logged in employee.

Employee (role-based)

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

Credentials	Employee's central user account and password.
Prerequisites	• The employee exists in the One Identity Manager database. • The central user account is entered in the employee's master data. • The system user password is entered in the employee's master data. • The employee 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	If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee's main identity is used for authentication. • If the parameter is not set, the employee's subidentity is used for authentication. A dynamic system user determined from the employee's application roles. The user interface and the write permissions are loaded through this system user. Changes to the data are assigned to the logged in employee.

Employee (dynamic)

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

Credentials	Employee's central user account and password.

Prerequisites	• The employee exists in the One Identity Manager database. • The central user account is entered in the employee's master data. • The system user password is entered in the employee's master data. • The configuration data for dynamically determining the system user is defined in the application. Thus, an employee 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	If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. The application configuration data is used to determine a system user, which is automatically assigned to the employee. The user interface and write permissions are loaded through the system user that is dynamically assigned to the logged in employee. Changes to the data are assigned to the logged in employee.

Related topics

- Configuration data for system user dynamic authentication on page 100

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	• The system user with permissions exists in the One Identity Manager database. • The employee exists in the One Identity Manager database. • Permitted logins are entered in the employee's master data. The logins are expected in the form: `domain\user`. • The system user is entered in the employee's master data.
Set as default	No
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	All employee logins saved in the One Identity Manager database are found. The employee whose login data matches that of the current user is used for logging in. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. The user interface and access permissions are loaded through the system user that is directly assigned to the employee 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	• The employee exists in the One Identity Manager database. • Permitted logins are entered in the employee's master data. The logins are expected in the form: `domain\user`.

	• The employee 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	All employee logins saved in the One Identity Manager database are found. The employee whose login data matches that of the current user is used for logging in. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. A dynamic system user determined from the employee's application roles. The user interface and the write permissions are loaded through this system user. Data modifications are attributed to the current user account.

Account based system user

Credentials	The authentication module uses the Active Directory login data of the user currently logged in on the workstation.
Prerequisites	• The system user with permissions exists in the One Identity Manager database. • Permitted logins are entered in the system user's master 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	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. The user interface and the write permissions are loaded through the system user. Data modifications are attributed to the current user account.

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	• The system user with permissions exists in the One Identity Manager database. • The employee exists in the One Identity Manager database. • The system user is entered in the employee's master data. • The Active Directory user account exists in the One Identity Manager database and the employee is entered in the user account's master data.
Set as default	Yes
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	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 determines which employee is assigned to the user account. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity

is used for authentication.

- If the parameter is not set, the employee’s subidentity is used for authentication.

The user interface and access permissions are loaded through the system user that is directly assigned to the employee found. If a system user is not assigned to the employee, the system user from the **SysConfig | Logon | DefaultUser** configuration parameter is used.

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.

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	• The employee exists in the One Identity Manager database. • The employee is assigned at least one application role. • The Active Directory user account exists in the One Identity Manager database and the employee is entered in the user account's master data.
Set as default	Yes
Single sign-on	Yes
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	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 determines which employee is assigned to the user account. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration

parameter controls which employee identity is used for authentication.

- If this configuration parameter is set, the employee’s main identity is used for authentication.
- If the parameter is not set, the employee’s subidentity is used for authentication.

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

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.

Active Directory user account (manual input)

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

Credentials	Login name and password for registering with Active Directory. You do not have to enter the domain.
Prerequisites	• The employee exists in the One Identity Manager database. • The Active Directory user account exists in the One Identity Manager database and the employee is entered in the user account's master data. • Domains permitted for login 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 employee 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	Yes

login allowed	
Remarks	The user‘s identity is determined from a predefined list of permitted Active Directory domains. The corresponding user account and employee are determined in the One Identity Manager database, which the user account is assigned to. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. The application configuration data is used to determine a system user, which is automatically assigned to the employee. The user interface and write permissions are loaded through the system user that is dynamically assigned to the logged in employee. 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 module Active Directory Module is installed.

Credentials	Login name and password for registering with Active Directory. You do not have to enter the domain.
Prerequisites	• The employee exists in the One Identity Manager database. • The employee is assigned at least one application role. • The Active Directory user account exists in the One Identity Manager database and the employee is entered in the user account's master data. • Domains permitted for login login are entered in the **TargetSystem | ADS | AuthenticationDomains** configuration parameter.
Set as default	Yes
Single sign-on	No
Front-end	Yes

login allowed	
Web Portal login allowed	Yes
Remarks	The user‵s identity is determined from a predefined list of permitted Active Directory domains. The corresponding user account and employee are determined in the One Identity Manager database, which the user account is assigned to. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. A dynamic system user determined from the employee's application roles. The user interface and the write 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 module 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	• The employee exists in the One Identity Manager database. • The Active Directory user account exists in the One Identity Manager database and the employee is entered in the user account's master data. • The configuration data for dynamically determining the system user is defined in the application. Thus, an employee can, for example, be assigned a system user dynamically depending on their department membership.
Set as default	No
Single sign-on	Yes
Front-end	Yes

login allowed	
Web Portal login allowed	Yes
Remarks	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 determines which employee is assigned to the user account. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. The application configuration data is used to determine a system user, which is automatically assigned to the employee. The user interface and write permissions are loaded through the system user that is dynamically assigned to the logged in employee. 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 100

LDAP user account (role-based)

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

Credentials	Login name, identifier, distinguished name or user ID of an LDAP user account. LDAP user account's password.
Prerequisites	• The employee exists in the One Identity Manager database. • The employee is assigned at least one application role. • The LDAP user account exists in the One Identity Manager database and the employee is entered in the user account's master data. • The configuration data for dynamically determining the system user is

	defined in the application. Thus, an employee 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	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. Domains permitted for login login are entered in the **TargetSystem | LDP| Authentication | RootDN** configuration parameter. If log in uses a distinguished name, the LDAP user account is determined that uses this distinguished name. One Identity Manager determines which employee is assigned to the LDAP user account. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. A dynamic system user determined from the employee's application roles. The user interface and the write permissions are loaded through this system user. Data modifications are attributed to the current user account.

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	The configuration parameter allows configuration of the LDAP authentication module.
TargetSystem | LDAP |	The configuration parameter specified the authentication mechanism. Permitted values are **Secure**, **Encryption**, **SecureSocketsLayer**,

Configuration parameter	Meaning
Authentication | Authentication	**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. The default is **ServerBind**.
TargetSystem | LDAP | Authentication | Port	LDAP server's port. The default is port **389**.

TargetSystem | LDAP | Authentication | RootDN	The configuration parameter contains a pipe (|) delimited list of root domains to use for finding the user account for authentication.
	Syntax:
	`DC=<MyDomain>|DC=<MyOtherDomain>`
	Example:
	`DC=Root1,DC=com|DC=Root2,DC=de`

TargetSystem | LDAP | Authentication | Server	The configuration parameter contains the name of the LDAP server.

LDAP user account (dynamic)

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

Credentials	Login name, identifier, distinguished name or user ID of an LDAP user account. LDAP user account's password.
Prerequisites	• The employee exists in the One Identity Manager database. • The LDAP user account exists in the One Identity Manager database and the employee is entered in the user account's master data. • The configuration data for dynamically determining the system user is defined in the application. Thus, an employee 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	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. Domains permitted for login login are entered in the **TargetSystem | LDP| Authentication | RootDN** configuration parameter. If log in uses a distinguished name, the LDAP user account is determined that uses this distinguished name. One Identity Manager determines which employee is assigned to the LDAP user account. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. The application configuration data is used to determine a system user, which is automatically assigned to the employee. The user interface and write permissions are loaded through the system user that is dynamically assigned to the logged in employee. Data modifications are attributed to the current user account.

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

Table 31: Configuration parameters for the authentication module

Configuration parameter	Meaning
TargetSystem | LDAP | Authentication	The configuration parameter allows configuration of the LDAP authentication module.
TargetSystem | LDAP | Authentication | Authentication	The configuration parameter specified the 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. The default is **ServerBind**.

Configuration parameter	Meaning
TargetSystem | LDAP | Authentication | Port	LDAP server's port. The default is port **389**.

TargetSystem | LDAP | Authentication | RootDN	The configuration parameter contains a pipe (|) delimited list of root domains to use for finding the user account for authentication.
	Syntax:
	`DC=<MyDomain>|DC=<MyOtherDomain>`
	Example:
	`DC=Root1,DC=com|DC=Root2,DC=de`

TargetSystem | LDAP | Authentication | Server	The configuration parameter contains the name of the LDAP server.

Related topics

- Configuration data for system user dynamic authentication on page 100

HTTP header

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

Credentials	Employee's central user account or personnel number.
Prerequisites	• The system user with permissions exists in the One Identity Manager database. • The employee exists in the One Identity Manager database. • The central user account or personnel number is entered in the employee's master data. • The system user is entered in the employee's master data.
Set as default	No
Single sign-on	Yes
Front-end	No

login allowed	
Web Portal login allowed	Yes
Remarks	You must pass the user (in the form: `UserName =<user name of authenticated user>`) in the HTTP header. The employee is found in the One Identity Manager database whose central user account or personnel number matches the user name passed down. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. The user interface and the write permissions are loaded through the system user that is directly assigned to the logged in employee. If a system user is not assigned to the employee, the system user from the **SysConfig | Logon | DefaultUser** configuration parameter is used. Changes to the data are assigned to the logged in employee.

HTTP header (role-based)

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

Credentials	Employee's central user account or personnel number.
Prerequisites	• The employee exists in the One Identity Manager database. • The central user account or personnel number is entered in the employee's master data. • The employee 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	You must pass the user (in the form: `UserName =<user name of authenticated user>`) in the HTTP header. The employee is found in the One Identity Manager database whose central user account or personnel number matches the user name passed down. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. A dynamic system user determined from the employee's application roles. The user interface and the write permissions are loaded through this system user. Changes to the data are assigned to the logged in employee.

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 detailed 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	• The system user with permissions exists in the One Identity Manager database. • The employee exists in the One Identity Manager database. • The system user is entered in the employee's master data. • The user account exists in the One Identity Manager database and the employee is entered in the user account's master data.
Set as default	No
Single sign-on	No

Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	One Identity Manager determines which employee is assigned to the user account. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. • If this configuration parameter is set, the employee’s main identity is used for authentication. • If the parameter is not set, the employee’s subidentity is used for authentication. The user interface and access permissions are loaded through the system user that is directly assigned to the employee found. 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.

Related topics

- OAuth 2.0 / OpenID Connect configuration on page 106
- Expiry of the OAuth 2.0/OpenID Connect authentication on page 106

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 detailed 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	• The employee exists in the One Identity Manager database.

	- The employee is assigned at least one application role. - The user account exists in the One Identity Manager database and the employee is entered in the user account's master data.
Set as default	No
Single sign-on	No
Front-end login allowed	Yes
Web Portal login allowed	Yes
Remarks	One Identity Manager determines which employee is assigned to the user account. If an employee has more than one identity, the **QER | Person | MasterIdentity | UseMasterForAuthentication** configuration parameter controls which employee identity is used for authentication. - If this configuration parameter is set, the employee’s main identity is used for authentication. - If the parameter is not set, the employee’s subidentity is used for authentication. A dynamic system user determined from the employee's application roles. The user interface and the write 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.

Related topics

- OAuth 2.0 / OpenID Connect configuration on page 106
- Expiry of the OAuth 2.0/OpenID Connect authentication on page 106

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 allowed	No
Web Portal login allowed	No
Remarks	You must not change the system user **sa**. The system user is overwritten with each schema update.

Web agent authentication module

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 allowed	No
Web Portal login allowed	No
Remarks	You must not change the system user **sa**. The system user is overwritten with each schema update.

Component authentication module

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 allowed	No
Web Portal login allowed	No
Remarks	You must not change the system user **sa**. The system user is overwritten with each schema update.

Crawler

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 allowed	No
Web Portal login allowed	No
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 access code or the employee's answers to the password questions. In the case of login with an access code, this information is deleted after a successful login.

Credentials	Central user account and access code. - 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	• The employee exists in the One Identity Manager database. • Using the central user account: The central user account is entered in the employee's master data. • Using the target system user account: The user account exists in the One Identity Manager database and the employee is entered in the master data of the employee’s user account. • The employee is not deactivated or has the certification status **New**. • The employee has an access code 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 detailed information about configuring Password Reset Portal, see the *One Identity Manager Web Application Configuration Guide*.

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 employee’s central user account is used.

Table 32: Configuration parameters for the authentication module

Configuration parameter	Meaning
QER | Person | PasswordResetAuthenticator | SearchTable	This configuration parameter contains the table in the One Identity Manager schema in which user information is stored. The table must contain a foreign key with the name `UID_Person`, which points to the `Person` table. Example: `ADSAccount`

Configuration parameter	Meaning
QER | Person | PasswordResetAuthenticator | SearchColumn	This configuration parameter contains the column from the One Identity Manager table (**SearchTable**), which is used to search for user name of the current user. Example: `CN`
QER | Person | PasswordResetAuthenticator | EnabledBy	This configuration parameter contains a pipe (|) delimited list of Boolean columns from the One Identity Manager (**SearchTable** table), which enables user account login.
QER | Person | PasswordResetAuthenticator | DisabledBy	This configuration parameter contains a pipe (|) delimited list of Boolean columns from the One Identity Manager table (**SearchTable**), which disables user account login. Example: `AccountDisabled`

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 access code or the employee’s answers to the password questions. In the case of login with an access code, this information is deleted after a successful login.

Credentials

Central user account and access code.

- 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

- The employee exists in the One Identity Manager database.
- Using the central user account: The central user account is entered in the employee's master data.
- Using the target system user account: The user account exists in the One Identity Manager database and the employee is entered in the master data of the employee’s user account.
- The employee is not deactivated or has the certification status **New**.
- The employee has an access code or the questions and answers for the password prompt have been specified.

	• The employee 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	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 detailed information about configuring Password Reset Portal, see the *One Identity Manager Web Application Configuration Guide*. A dynamic system user determined from the employee's application roles. The user interface and the write permissions are loaded through this system user.

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 employee’s central user account is used.

Table 33: Configuration parameters for the authentication module

Configuration parameter	Meaning
QER | Person | PasswordResetAuthenticator | SearchTable	This configuration parameter contains the table in the One Identity Manager schema in which user information is stored. The table must contain a foreign key with the name `UID_Person`, which points to the `Person` table. Example: `ADSAccount`
QER | Person | PasswordResetAuthenticator | SearchColumn	This configuration parameter contains the column from the One Identity Manager table (**SearchTable**), which is used to search for user name of the current user. Example: `CN`
QER | Person | PasswordResetAuthenticator | EnabledBy	This configuration parameter contains a pipe (|) delimited list of Boolean columns from the One Identity Manager (**SearchTable** table), which enables user account login.
QER | Person | PasswordResetAuthenticator | DisabledBy	This configuration parameter contains a pipe (|) delimited list of Boolean columns from the One Identity Manager table (**SearchTable**), which disables user account login. Example: `AccountDisabled`

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 94
- Assigning authentication modules to applications on page 95
- Disabling or enabling authentication modules for applications on page 95
- Authentication module properties on page 96
- Initial data for authentication modules on page 96
- Configuration data for system user dynamic authentication on page 100
- One Identity Manager authentication modules on page 67
- Managing permissions to program features on page 60

Enabling authentication modules

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. Save the changes.

Related topics

- Disabling or enabling authentication modules for applications on page 95

Assigning authentication modules to applications

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

> 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 assign an authentication module to applications

1. In the Designer, select the **Base data | Security settings | Authentication modules** category.
2. Select **View | Select table relations** and enable the `DialogProductHasAuthentifier` table.
3. In List Editor, select the authentication module.
4. Assign the application in the **Applications** edit view.
5. Save the changes.

Related topics

- Disabling or enabling authentication modules for applications on page 95

Disabling or enabling authentication modules for applications

To disable an authentication module for an application

1. In the Designer, select the **Base data | Security settings | Programs** category.
2. In List Editor, select the application and click on **Usage overview**.
3. In the **Effective authenticators** form element, select the authentication module.
4. Use the **Edit object** task to start the Object Editor.
5. In the **Disabled** property, set the value to **True**.
6. Save the changes.

To enable an authentication module for an application

1. In the Designer, select the **Base data | Security settings | Programs** category.
2. In List Editor, select the application and click on **Usage overview**.
3. In the **Disabled authenticators** form element, select the authentication module.
4. Use the **Edit object** task to start the Object Editor.

5. In the **Disabled** property, set the value to **False**.
6. Save the changes.

Related topics

- Assigning authentication modules to applications on page 95
- Enabling authentication modules on page 94

Authentication module properties

Table 34: Authentication module properties

Property	Meaning
Enabled	Specifies whether the authentication module can be used.
Display name	This name is used to identify the authentication module in the administration tool's login window.
Authentication module	Internal name of the authentication module.
Authentication type	Specifies the type of authentication module. 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.
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.

Initial data for authentication modules

The initial data is one part of the authentication string (parameter-value pair without module ID). Initial data from the authentication string is preallocated by default for each authentication instance.

The authentication string is formatted as follows:

```
Module=<name>;<property1>=<value1>;<property2>=<value2>,…
```

Example:

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

To specify initial data

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>
   ```

You can use different initial data depending on the authentication module.

Table 35: Initial data for authentication modules

Module Display Name	Authentication module	Parameters	Meaning/Comment
System users	DialogUser	User	User name.
		Password	User password.
Active Directory user account	ADSAccount		
Active Directory user account (dynamic)	DynamicADSAccount	Product	Use case. The system user is determined through the use case configuration data.
Active Directory user account (manual input)	DynamicManualADS	Product	Use case. 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.

Module Display Name	Authentication module	Parameters	Meaning/Comment
		Password	User password.
Active Directory user account (role-based)	RoleBasedADSAccount		No parameters required
Active Directory user account (manual input/role-based)	RoleBasedManualADS	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	User password.
Employee	Employee	User	Employee's central user account.
		Password	User password.
Employee (dynamic)	DynamicPerson	Product	Use case. The system user is determined through the use case configuration data.
		User	User name.
		Password	User password.
Employee (role-based)	RoleBasedPerson	User	User name.
		Password	User password.
HTTP header	HTTPHeader	Header	HTTP Header to use.
		KeyColumn	Comma delimited list of key columns in the `Person` table to be searched for user names. Default: `CentralAccount, PersonnelNumber`

Module Display Name	Authentication module	Parameters	Meaning/Comment
HTTP header (role-based)	RoleBasedHTTPHeader		HTTP header to use.
		KeyColumn	Comma delimited list of key columns in the `Person` table to be searched for user names. Default: `CentralAccount`, `PersonnelNumber`
LDAP user account (dynamic)	DynamicLdap	User	User name. Default: `CN`, `DistinguishedName`, `UserID`, `UIDLDAP`
		Password	User password.
LDAP user account (role-based)	RoleBasedLdap	User	User name. Default: `CN`, `DistinguishedName`, `UserID`, `UIDLDAP`
		Password	User password.
Generic single sign-on (role-based)	RoleBasedGeneric	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.
		SearchColumn	Column from the `SearchTable` in which to search for the user name of the logged-in user.
		DisabledBy	Pipe (|) delimited list of Boolean columns which block a user account from logging in.
		EnabledBy	Pipe (|) delimited list of Boolean columns which release a user account for logging in.
OAuth 2.0/OpenID Connect	OAuth		Dependent on the authentication method of the secure token service.
OAuth	OAuthRoleBased		Dependent on the authen-

Module Display Name	Authentication module	Parameters	Meaning/Comment
2.0/OpenID Connect (role-based)			tication method of the secure token service.
Account based system user	DialogUserAccountBased		No parameters required
User account	QERAccount		No parameters required
User account (role-based)	RoleBasedQERAccount		No parameters required
Password reset	PasswordReset		No parameters required
Password reset (role-based)	RoleBasedPasswordReset		No parameters required

Related topics

- Configuration data for system user dynamic authentication on page 100
- One Identity Manager authentication modules on page 67

Configuration data for system user dynamic authentication

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

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 employee 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 rights and user interface structures. The function groups allow you to map the functions an employee 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 employee 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 102
- Example of a system user assignment using a selection criterion on page 103
- Example of a function group assignment on page 103
- Granting One Identity Manager schema permissions through permissions groups on page 33

Example of a simple system user assignment

All employees 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 100
- Example of a system user assignment using a selection criterion on page 103
- Example of a function group assignment on page 103
- Granting One Identity Manager schema permissions through permissions groups on page 33

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 employee'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 access 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 employees use the **dlg_all** system user to log in.

Change the configuration data as follows:

```
<DialogUserDetect>
      <Usermappings>
            <Usermapping
                  DialogUser = "dlg_kst"
                  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 100
- Example of a simple system user assignment on page 102
- Example of a function group assignment on page 103
- Granting One Identity Manager schema permissions through permissions groups on page 33

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 employees 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 100
- Example of a simple system user assignment on page 102
- Example of a system user assignment using a selection criterion on page 103
- Granting One Identity Manager schema permissions through permissions groups on page 33

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 when the next permissions based action on the connection at 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 configuration

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 detailed information about the authorization code flow, see, for example, the OAuth Specification or the OpenID Connect Specification.

To use the 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 106
- Creating the OAuth 2.0/OpenID Connect configuration on page 108
- Assigning OAuth 2.0/OpenID Connect configuration to web applications on page 112
- Defining enabling and disabling columns for the determination of user accounts on page 114
- OAuth 2.0/OpenID Connect on page 86
- OAuth 2.0/OpenID Connect (role-based) on page 87
- Logging information about OAuth 2.0/OpenID Connect authentication on page 114

Expiry of the OAuth 2.0/OpenID Connect authentication

At the endpoint of the authorization, the web application (or native application) requests the authorization code. 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 (`QBMIdentityClient` table)
    a. Certificate text (`QBMIdentityClient.CertificateText`).
    b. Subject or thumbprint from the local memory (`QBMIdentityClient.CertificateSubject` and `QBMIdentityClient.CertificateThumbPrint`).
    c. Certificate endpoint (`QBMIdentityClient.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 (`QBMIdentityProvider` table)
    a. Certificate text ((`QBMIdentityProvider.CertificateText`).
    b. Subject or thumbprint from the local memory (`QBMIdentityProvider.CertificateSubject` and `QBMIdentityProvider.CertificateThumbPrint`).
    c. Certificate endpoint (`QBMIdentityProvider.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 (`QBMIdentityProvider.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 108
- OAuth 2.0/OpenID Connect on page 86
- OAuth 2.0/OpenID Connect (role-based) on page 87

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:
        a. Select **Automatic configuration data discovery**.
        b. Enter the address (URL) for automatic determination of the configuration data in the input field, or select an example address through the selection menu.
        c. Click **Execute**.
        d. The configuration data is determined and a dialog window is displayed. To accept the configuration data, click **OK**.
    - 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 36: 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

Property	Description
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
UserInfo endpoint	URL of the OpenID Connect UserInfo endpoint.
Self-signed certificates allowed	Specifies whether self-signed certificates are allowed for connecting to the token endpoint and UserInfo endpoint.
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.
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.

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 37: 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	Subject of the certificate used for verification. The subject or

Property	Description
certificate	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 38: Determining the login information

Property	Description
Value for the search	Full name of the claim type from which the login information is determined on the identity provider. Example: name of an entity http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier If you have determined the configuration data automatically, select a value from the list.
Column to search	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 `UID_Person`, which points to the `Person` table. Example: `ADSAccount.ObjectGUID`
User name value	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 (`XUserInserted` and `XUserUpdated` columns). Example: User Principle Name (UPN) http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn If you have determined the configuration data automatically, select a value from the list.

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 input field.

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

Table 39: General information about the application

Property	Description
Display name	Display name of the application.
Description	Text field for additional explanation.
Client ID	ID of the application on the identity provider. For native 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
Default	Specifies whether this is a standard application for native applications.

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

Table 40: 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 41: Information about the application certificate

Property	Description
Authentication method	Authentication method at the token endpoint. Permitted values are: • **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. Encryption with the private key is also applied.
Token endpoint certificate	Hexadecimal thumbprint of the certificate for validating the token.

15. To create the identity provider and the application in the One Identity Manager database, click **Next**.
16. Click **Finish** to complete the wizard.

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. Save the changes.

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 more 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 108
- Defining enabling and disabling columns for the determination of user accounts on page 114

Defining enabling and disabling columns for the determination of user accounts

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.

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 offered.

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 logon.
5. Save the changes.

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 logon.
5. Save the changes.

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.

Multi-factor authentication in One Identity Manager

Table 42: Multi-factor authentication configuration parameters

Configuration parameter	Meaning
QER | Person | Defender	This configuration parameter specifies whether classic Starling Two-Factor Authentication integration is supported.
QER | Person | Defender | ApiEndpoint	This configuration parameter contains the URL of the Starling 2FA API end point used to register new users.
QER | Person | Defender | ApiKey	This configuration parameter contains your company's subscription key for accessing the Starling Two-Factor Authentication interface.
QER | Person | Starling	This configuration parameter specifies whether One Identity Starling Cloud is supported. Initiate your subscription within your One Identity on-prem product and join your on-prem solutions to our One Identity Starling Cloud platform. Giving your organization immediate access to a number of cloud-delivered microservices, which expand the capabilities of your One Identity on-prem solutions. We will continuously make available new products and features to our Starling Cloud platform. For a free trial of our One Identity Starling offerings and to get the latest product feature updates, visit cloud.oneidentity.com.
QER | Person | Starling | ApiEndpoint	This configuration parameter contains the touch endpoint for login to the One Identity Starling software-as-a-service platform. The value is determined by the Starling configuration wizard.
QER | Person | Starling | ApiKey	This configuration parameter contains the credential string for login to the One Identity Starling software-as-a-service platform. The value is determined by the Starling configuration wizard.

You can set up multi-factor authentication 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.

Use One Identity Manager One Identity Starling Two-Factor Authentication for multi-factor authentication. This service is normally provided over a One Identity Starling Cloud platform. If your company does not use a Starling Cloud, select the conventional Starling Two-Factor Authentication integration. Use configuration parameters to specify which of the two solutions are applied in your company.

To be able to use multi-factor authentication

1. Register your company in Starling Two-Factor Authentication.

   For more detailed information, see the Starling Two-Factor Authentication documentation.

2. Specify which authentication solution is used.

   - To use Starling Cloud

     a. Start the Launchpad.

     b. Select **Connection to Starling Cloud** and click **Run**.

        This starts the Starling Cloud configuration wizard.

     c. Follow the Starling Cloud configuration wizard’s instruction.

     The configuration parameters under **QER | Person | Starling** are enabled and the authentication information is entered.

   - To use conventional Starling Two-Factor Authentication integration

     a. In the Designer, enable the **QER | Person | Defender** configuration parameter.

        - Enable the **QER | Person | Defender | ApiKey** configuration parameter and enter your company’s subscription key as the value for accessing the Starling Two-Factor Authentication interface.

        The default URL of the Starling 2FA API end point is already entered in the **QER | Person | Defender | ApiEndpoint** configuration parameter.

3. Enable assigning by event for the `PersonHasQERResource` table. For more information, see Editing table properties on page 118.

4. (Optional) Specify whether the security code must be requested from the Starling 2FA app. For more information, see Requesting a security code on page 119.

5. (Optional) Specify whether requests can be approved using the Starling 2FA app. For more information, see Allowing approval decisions using the Starling 2FA app on page 120.

6. In the Manager, enable the **New Starling 2FA token** service item. For more information, see Preparing the Starling 2FA token request on page 118.

If the user's telephone number has changed, cancel the current Starling 2FA token and request a new one. If the Starling 2FA token is no longer required, cancel it anyway.

For detailed information, see the following guides:

Theme	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
Setting up Starling Two-Factor Authentication in the web project	One Identity Manager Web Application Configuration Guide
Requesting the Starling 2FA Token Requesting products requiring multi-factor authentication Approving requests with multi-factor authentication Attestation with multi-factor authentication	One Identity Manager Web Portal User Guide

Editing table properties

> NOTE: If the **Assign by event** option is enabled, the `HandleObjectComponent` process is placed in the job queue as soon as a resource assignment is added to or removed from an employee.

To enable assigning by event for a table

1. In the Designer, select **One Identity Manager Schema**.
2. Select the `PersonHasQERResource` table and start Schema Editor using the **Show table definition** task.
3. In the **Table properties** view, select the **Table** tab and enable the **Assign by event** option.
4. Save the changes.

For detailed information about editing table definitions, see the *One Identity Manager Configuration Guide*.

Preparing the Starling 2FA token request

One Identity Manager users must be registered with Starling Two-Factor Authentication in order to use multi-factor authentication. To register, a user must request the Web Portal Token in the Starling 2FA. Once the request has been granted approval, the user receives a link to the Starling Two-Factor Authentication app and a Starling 2FA user ID. The app generates one-time passwords, which are required for authentication. The Starling 2FA user ID is saved in the user's employee master data.

> NOTE: The user's default email address, mobile phone and country must be stored in their master data. This data is required for registering.

To facilitate requesting a Starling 2FA token

1. Select the **IT Shop | Service catalog | Predefined** category.
2. Select **New Starling 2FA token** in the results list.
3. Select the **Change master data** task.
4. Disable **Not available**.
5. Save the changes.

The Starling 2FA token request must be granted approval by the request recipient's manager.

Requesting a security code

Table 43: Configuration parameter for requesting Starling 2FA security codes

Configuration parameter	Meaning
QER | Person | Defender | DisableForceParameter QER | Person | Starling | DisableForceParameter	The configuration parameters specify whether Starling 2FA is forced to send the security code by SMS or phone call if one of these options is selected for multi-factor authentication. If the configuration parameters are enabled, Starling 2FA can refuse this request; the user must then request the security code by the Starling 2FA app.

If the security code is requested for an attestation, request, or request approval, the user decides how the security code is sent. The following options are available:

- By Starling 2FA app
- By SMS
- By phone call

By default, Starling 2FA is forced to send the security code by SMS or by phone call if the user has selected one of these options. However, for security reasons, the user should use the Starling 2FA app to generate the security code. If the app is installed on the user's mobile phone, Starling 2FA can refuse the SMS or phone demand and the user must generate the security code using the app.

To use this method

- If you use Starling Cloud, in the Designer, set the **QER | Person | Starling | DisableForceParameter** configuration parameter.

  - OR -

- If you use classic Starling Two-Factor Authentication integration, in the Designer, set the **QER | Person | Defender | DisableForceParameter** configuration parameter.

  Starling 2FA can refuse to transmit the security code by SMS or phone call if the Starling 2FA app is installed on the phone. Then the security code must be generated by the app.

If the configuration parameter is not set (default), Starling 2FA is forced to send the security code by SMS or phone call.

Allowing approval decisions using the Starling 2FA app

Table 44: Configuration parameters for approving with the Starling 2FA app

Configuration parameter	Meaning
QER | Person | Starling | UseApprovalAnywhere	This configuration parameter defines whether requests can be approved by Starling 2FA app.
QER | Person | Starling | UseApprovalAnywhere | SecondsToExpire	This configuration parameter specifies the timeout in seconds after which approval by Starling 2FA app expires.

To provide approvers who are temporarily unable to access One Identity Manager tools, with the option of making approval decisions for requests, you can set up approval by Starling 2FA app. This means, approvers are prompted by the Starling 2FA app to approve or deny a request . This option is only available if you use Starling Cloud for multi-factor authentication and the approvers are registered with Starling Two-Factor Authentication.

To use the Starling 2FA app for approval decisions

- In the Designer, set the **QER | Person | Starling | UseApprovalAnywhere** configuration parameter.

  The approver must make the approval decision within 5 minutes. If this times out, the approver must use the Web Portal to approve the request .

  To change the timeout, modify the value in the **QER | Person | Starling | UseApprovalAnywhere | SecondsToExpire** configuration parameter. Enter a timeout in seconds.

Granulated permissions for the SQL Server and database

To implement a One Identity Manager database or a One Identity Manager History Database on a SQL Server or a managed instance in Azure SQL Database, you are provided with SQL Server logins and database users for 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. It may be necessary to set up an additional database user to use a One Identity Manager History Database.

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

Related topics

- Minimum access levels of One Identity Manager tools on page 121
- Displaying database server logins on page 123
- Displaying users' access levels on page 124
- Displaying server and database permissions on page 124

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 45: 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)
AppServer.Installer.CMD.exe	Configuration user
API Designer	Configuration user
API Server	End user
Configuration Wizard	Administrative user
Crypto Configuration	Configuration user
Data Import	End user Configuration user (saves import definition)
DataImporterCMD.exe	End user
Database Compiler	Configuration user
DBCompilerCMD.exe	Configuration user
Database Transporter	Configuration user
DBTransporterCMD.exe	Administrative user
DBClone	Administrative user
DBComparer	Configuration user
Designer	Configuration user
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, consistency checking or opening target systems' synchronization projects.
HistoryDB Manager	End user
Object Browser	End user
One Identity Manager Service	Configuration users for process collection with the

Tool	Minimum access level
	MSSQLJobProvider
Report Editor	Configuration user
Schema Extension	Configuration user
SchemaExtensionCmd.exe	Configuration user
Server Installer	Configuration user
Software Loader	Configuration user
SoftwareLoaderCMD.exe	Configuration user
Synchronization Editor	Configuration user
System Debugger	Configuration user
Web Designer	Configuration user
Web Designer Configuration Editor	Configuration user
VI.WebDesigner.CompilerCmd.exe	Configuration user
WebDesigner.InstallerCMD.exe	Configuration user
Web Portal	End user
Password Reset Portal	End user
Operations Support Web Portal	End user
Quantum.MigratorCmd.exe	Administrative user

Related topics

- Granulated permissions for the SQL Server and database on page 121

Displaying database server logins

To display login information

1. In the Designer, select the **Base data | Security settings | Database server permissions | Database server login** category.
2. Select the database server login. 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 the database roles and server roles that are assigned, select the **Database or server role** tab.

Displaying users' access levels

> NOTE:
>
> - If you select an existing database connection in the login 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 server logins on page 123

Displaying server and database permissions

Server and database permissions are predefined and cannot be modified.

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

To display server and database permissions

- In the Designer, in the **Base data | Security settings | Database server permissions | Database server login** category, select the server or database role.

  This opens the List Editor showing a list of permissions.

About us

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

Contacting us

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

Technical support resources

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

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

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

Index