Axidian Access

Technical Documentation

Version: 8.2

Date: 2026-08-14

Contents

About Axidian Access

Axidian Access allows you to build a centralized access management system for company information resources.

Axidian Access includes various modules for integration with target systems using technologies such as SAML, OpenID Connect, RADIUS, ADFS, Enterprise SSO, and authentication agents.

This approach allows you to protect access to all information systems of the enterprise and adapt the solution to the needs of the customer.

Axidian Access integration modules

Authentication and access management

Axidian Access not only makes it possible to use reliable user authentication technologies, but also implements access control functions such as SSO and access control for different user groups.

Different authentication methods can be combined into chains to create multifactor authentication (MFA). The architecture of the solution allows you to quickly add support for new technologies.

Authentication technologies supported in Axidian Access

Installation

Axidian Access uses an on-premise deployment scheme — directly in your work environment or in your own data center. You fully control the operating environment of the user authentication system, the data of which is stored locally and not transferred to cloud storage.

Axidian Access supports capacity scaling — installation in cluster mode. This ensures the required levels of fault tolerance and performance and does not require the purchase of additional licenses.

Base components

The platform is based on the base modules that implement the business logic of the system and ensure the functioning of the server infrastructure and the management tools.

Core Server authentication and management server

The server is the core of the system and ensures the functioning of the entire platform; it performs user authentication and implements the business logic of the solution. The server is an ASP.NET application and supports installation in cluster mode, allowing a high level of performance and fault tolerance regardless of the implementation scale.

Install and configure Core Server

User catalog

Axidian Access uses end user data from Active Directory. These are the users who go through the authentication procedure using Axidian Access.

Prepare the user catalog

Databases

All system data is stored in a single storage, which is directly accessed only by the server. Data is stored and transmitted to or from the server in encrypted form. The storage can be located either in Microsoft SQL or in PostgreSQL.

Create and configure the data storage

Management Console

It is designed in the format of a web application in which Axidian Access administrators can view and modify system parameters and user settings, as well as view the system log.

Install and configure Management Console

User Console

It allows you to register or change authentication data (smart cards, one-time password generators, fingerprints, and others), as well as view your login history.

Install and configure User Console

Integration modules

ADFS Extension

Two-factor authentication using ADFS

Enterprise Single Sign-On

Single Sign-On for legacy applications

Identity Provider

Multifactor authentication and single sign-on

IIS Extension

Authentication in applications based on Internet Information Services

NPS RADIUS Extension

Two-factor authentication for the RADIUS protocol

RDP Windows Logon

Two-factor authentication for the Remote Desktop Protocol

Windows Logon

Access to Windows with strong authentication

ADFS Extension

Web applications based on the Internet Information Services (IIS) server can be integrated with Axidian Access using the ADFS mechanism and the ADFS Extension component.

The ADFS Extension component implements a multifactor authentication provider for the Microsoft ADFS server, adding a second factor to the access gaining process. This approach makes it possible to integrate with target applications without modifying them: when logging in to an application, the user is redirected to the ADFS authentication web page, where the second factor is requested through the ADFS Extension authentication provider; after successful authentication, the user is returned to the target application.

Install and configure ADFS Extension

Authentication technologies

As the second factor, ADFS Extension supports authentication with the following technologies:

- OATH one-time passwords: TOTP and HOTP
- one-time codes sent by SMS and email
- one-time codes and push notifications in the Axidian Key mobile application
- a self-defined password

Where it can be used

The ADFS technology is supported by Microsoft web applications such as Outlook Web Access, SharePoint, Skype for Business, and others.

Enterprise Single Sign-On

Enterprise Single Sign-On (Enterprise SSO) implements the Single Sign-On approach for legacy applications that do not support SSO mechanisms. The system stores the user passwords for all applications that require credentials centrally and substitutes them into screen forms automatically when the application requires it.

Enterprise SSO relieves employees of memorizing passwords and keeping them secret, entering passwords manually from the keyboard, and changing passwords periodically according to password security policies.

For this purpose, the Enterprise SSO agent is installed on the user workstation. It tracks the start of applications and intercepts authentication forms when they appear on the screen. The agent also includes extensions for popular browsers (Internet Explorer, Google Chrome, Mozilla Firefox), which allows working with web applications.

Install and configure Enterprise SSO

Add and configure integrated applications

Where it can be used

The Enterprise SSO technology can be applied to any type of application (Windows, Java, Web, .NET) regardless of the architecture: single-tier, two-tier, three-tier, thick client, thin client, terminal applications.

Authentication technologies

Enterprise SSO supports the following authentication technologies:

- TOTP and HOTP one-time passwords
- one-time codes sent by SMS and email
- one-time codes and push notifications in the Axidian Key mobile application
- smart cards and USB tokens

How Enterprise SSO integrates with target systems

Enterprise SSO can be configured to work with an application without interfering with the server or the client part of that application. Support for a new application implies creating a special template in the XML format, implemented in the internal script-type language of Enterprise SSO. The language allows you to specify which application forms a reaction must be defined for. The reaction of Enterprise SSO may include repeated strong user authentication, filling in fields with registration data (such as the login and password), and activating the required controls (such as clicking the login button).

Password change in the target application

To minimize information security risks, most information systems support the ability to require the user to change the password value immediately after the first login to the system, or after the specified password lifetime expires. Enterprise SSO handles this situation and allows blocking the user access to the password change window automatically (transparently for the user), generating a new value, filling in the *New value* and *Confirmation* form fields, and clicking *OK*. After receiving a notification from the target system about the successful password change, the Enterprise SSO agent saves the new value in the Axidian Access database. From that moment, neither the user nor the administrator knows the new password value and therefore cannot log in to the target system bypassing Enterprise SSO.

The ability to handle the password change situation appears only if the ESSO application template supports a reaction to this type of window.

Terminal environment support

Enterprise SSO is adapted to work in a terminal environment, allowing employees to avoid the explicit use of their passwords when working with an application inside a terminal session. For this, the Enterprise SSO agent must be installed on the terminal server.

In some situations, when accessing especially critical applications, an employee may be required to go through an additional authentication procedure. If the technology implies the use of external equipment connected to the employee PC (such as a fingerprint scanner), communication occurs between the Enterprise SSO agent of the terminal server and the equipment. Enterprise SSO communicates over the Microsoft RDP or Citrix ICA protocols. This means that no additional

software installation is required on the employee PC, except for the driver and the set of run-time libraries required for the equipment to work.

Identity Provider

To organize multifactor authentication and single sign-on to web applications (Web Single Sign-On), the Identity Provider (IDP) module is used. For integration with target solutions, this module supports the following protocols:

- SAML 2.0 (Security Assertion Markup Language)
- OIDC 1.0 (OpenID Connect)
- OAuth 2.0

This guarantees compatibility with a wide range of commercial systems. Using IDP relieves the user of the need to memorize multiple credentials: only one set of credentials is required to access all integrated systems. Authentication is performed centrally on the Identity Provider side.

IDP is implemented in the format of a web application and is deployed in the customer infrastructure. During the access gaining process, the target application redirects the user to the IDP page for authentication; if it is successful, the user is redirected back to the target application with the *Authenticated* attribute, where their session is opened.

Install and configure Identity Provider

Authentication technologies

IDP supports the following user authentication technologies:

- the domain password
- OATH one-time passwords: TOTP and HOTP
- one-time codes sent by SMS and email
- one-time codes and push notifications in the Telegram messenger
- one-time codes and push notifications in the Axidian Key mobile application

Where it can be used

Both corporate on-premise applications (such as solutions from SAP, Citrix, and others) and cloud services such as Office 365, Salesforce, Slack, Google Workspace (formerly G Suite), and others can be included in the WebSSO and MFA perimeter.

The integration is performed on the server side, which makes it possible to use the MFA and WebSSO approach on any device with a browser: a PC, a smartphone, or a tablet.

IIS Extension

For authentication in web applications that use Internet Information Services (IIS) and do not support the ADFS mechanism, the specialized IIS Extension integration module has been developed.

IIS Extension is installed on the web server where the target application is deployed and allows two-factor authentication to be provided without interfering with its program code. The module intercepts authentication attempts, and after the username and password are entered, the user is redirected to a separate page where they must confirm the login with a one-time password.

The one-factor authentication mode is also supported. This mode is in demand for the Exchange ActiveSync (EAS) application and allows the domain password to be excluded from the authentication scheme. In this case, a non-domain password is used to access EAS, which is essentially a so-called application password used only for EAS. This password is entered by the user in the mobile client to access corporate mail.

Install and configure IIS Extension

Authentication technologies

IIS Extension supports the following user authentication technologies:

- OATH one-time passwords: TOTP and HOTP
- one-time codes sent by SMS and email
- push notifications in the Axidian Key mobile application
- one-time codes and push notifications in the Telegram messenger
- a self-defined password

Where it can be used

IIS Extension can be used for any IIS-based web applications, such as Outlook Web Access, RD Web Access, Exchange ActiveSync, and others.

NPS RADIUS Extension

NPS RADIUS Extension (RADIUS Extension) is an extension module for Microsoft Network Policy Server (NPS is included in Windows Server) that allows you to implement two-factor authentication for RADIUS-compatible services and applications.

For this, the following is required:

- Deploy an NPS server in the enterprise network that provides the ability to authenticate over the RADIUS protocol using the credentials of Active Directory catalog users.
- Configure the target application to authenticate users over the RADIUS protocol on the NPS server.
- Install RADIUS Extension on the NPS server, which processes authentication requests and requires the second authentication factor from the user.

Authentication by the second factor is performed on the Axidian Access server, and the verification result is transmitted through the NPS server to the target application.

Install and configure NPS RADIUS Extension

Authentication technologies

As the second factor, RADIUS Extension supports authentication with the following technologies:

- OATH one-time passwords: TOTP and HOTP
- one-time codes sent by SMS and email
- one-time passwords in the Telegram application
- push notifications with login confirmation in the Axidian Key mobile application

Where it can be used

Authentication over the RADIUS protocol can be used in many VPN and VDI solutions, for example in the software products of Cisco, Citrix, Check Point, and VMware.

RDP Windows Logon

The RDP Windows Logon module is used to implement two-factor authentication for RDP connections. In this case, the first factor is the domain password, and the second one is a one-time password or a login confirmation in the Axidian Key mobile application. The one-time password can be generated on the user side in the smartphone application, with a special key fob (an OTP token), or sent to the user by SMS or email.

RDP Windows Logon must be installed on the target terminal server the user logs in to. No components need to be installed on the client computer. The configuration with Remote Desktop Gateway is supported.

Install and configure RDP Windows Logon

Authentication technologies

RDP Windows Logon supports the following user authentication technologies:

- TOTP and HOTP one-time passwords
- one-time codes sent by SMS and email
- push notifications in the Axidian Key mobile application
- a self-defined password

Windows Logon

Windows Logon provides the ability to access Windows using strong authentication technologies in the Microsoft Active Directory environment. For this, the Windows Logon agent is installed on user workstations.

The agent installer is implemented as a standard MSI (Microsoft Windows Installer) package. This allows you to install and update the system in bulk quickly, using various tools such as Active Directory group policies, Microsoft System Center Configuration Manager (SCCM), and others.

For integration with the Windows operating system, the standard Credentials Provider mechanism for implementing a custom user authentication interface is used. This technology allows third-party developers to integrate their own authentication technologies with the Windows interface. It makes it possible not only to log in to Windows with Axidian Access technology, but also to authenticate with Axidian Access inside the operating system, for example when accessing domain resources or web applications.

Windows Logon supports all the authentication technologies available in Axidian Access — smart cards, RFID cards, one-time passwords.

Install and configure Windows Logon

Authentication technologies

Windows Logon supports the following authentication technologies:

- the domain password
- TOTP and HOTP one-time passwords
- one-time codes sent by SMS and email
- one-time codes and push notifications in the Axidian Key mobile application
- smart cards and USB tokens

Offline mode

To improve fault tolerance, Windows Logon can create a local cache on the user computer. Such a cache contains authentication data and is used when there is no connection to the server infrastructure (the offline mode), for example due to a connection loss or during a business trip.

The lifetime of the local cache can be limited by a number of days or a calendar date. The cache is created only for the users for whom this is explicitly allowed by the Axidian Access administrator. The Windows Data Protection API technology is used to protect the local data.

Managing Active Directory user passwords

Axidian Access does not replace the standard Active Directory authentication system, but automates the management of user passwords. In such a configuration, password authentication becomes an internal mechanism used only at the software level.

The Axidian Access administrator can configure the system so that at the moment the first authenticator is registered for a user, their password is automatically changed to a random value that is disclosed neither to the user nor to the system administrator. Thus, access to the domain becomes possible only with Windows Logon. Subsequently, the user password is changed automatically, either at the request of the operating system or on a specified schedule.

Authentication technologies

Authentication providers enable Axidian Access to work with user authentication technologies.

An authentication provider gives the system a unified interface for performing operations with a specific authentication technology: obtaining authentication data from the user for storage, and verifying the data.

Install and configure authentication providers

Axidian Access supports the following authentication technologies:

- Cryptographic smart cards and USB tokens, such as Rutoken, eToken, JaCarta, and others.
- RFID contactless cards (used as passes in access control systems) of the EM-Marin, HID iClass, HID Proximity, and Mifare formats.
- Hardware and software tokens generating one-time passwords over the OATH, TOTP, and HOTP protocols.
- One-time codes sent by SMS or email.
- Authentication with the mobile application and push notifications based on the Axidian Key application.

The Axidian Key mobile application is available for the iOS and Android operating systems. Using Axidian Key, the user confirms the login operation in the application on their smartphone. The operation details are displayed on the smartphone screen, where the user can make sure which system exactly is being logged in to. In addition, Axidian Key supports the generation of one-time passwords over the TOTP protocol.

Different technologies can be combined into one authentication method, thus implementing multifactor authentication (MFA).

Access policies

A policy is a set of settings of base components, integration modules, business applications, and accounts that can be applied to a specific selection of users from the user catalog. All users included in the policy receive accounts for applications with the corresponding settings, or the settings themselves from the policy.

A policy can define the following:

- The scope: the users who fall under the rules specified in this policy.
- The roles: administrator, operator, and inspector.
- The set and settings of the available authentication methods.
- The set of available business applications and their role accounts.
- The policy priority.

If a user falls within the scope of more than one policy, they get access to all applications added to these policies.

If the same application occurs more than once in the policies whose scope the user falls into, the user is assigned the access settings for this application from the policy with the highest priority.

For example, if policy #1 specifies the use of SMS OTP for ADFS Extension, and the higher-priority policy #2 specifies the use of Email OTP for ADFS Extension, the user must log in through ADFS using Email OTP.

If the policy with the highest priority has no settings for the application, the settings from the next policy by priority are used.

In ESSO scenarios, there may be exceptions related to the implementation specifics of ESSO Agent.

Licensing

Axidian Access products use the CAL (Client Access License) licensing scheme. Such a license grants the right to use Axidian Access products for one user. The number of Axidian Access user accounts is counted, while the number of client software installations is not limited.

Licenses can be purchased additionally. If necessary, you can revoke a license from some employees and allocate it to others.

How to manage licenses in Management Console

Axidian Access licenses are divided into two types:

1. The main Axidian Access license.
2. Licenses for additional modules.

Main license

This is the base license. It grants the right to register a user in the system, store their authentication information in the database, and use the basic functions — the administrator console, the user console, and the event log. Purchasing this license is required in any case, regardless of the required functionality of the solution.

Licenses for additional modules

The license allows you to use a specific module. Licenses are required for all integration modules. All business applications of a module work under the module license; separate licenses are not required for them.

A module license contains an allowed number of its uses. After a license is registered, its objects can be issued to users automatically; this happens during the authentication process.

The revocation of a license for a specific user also happens automatically, by means of a regularly recurring task, if the user in the catalog has been blocked or has lost the rights to use the module.

- *Axidian Access Windows Logon license*. Grants the right to use the Windows Logon module for strong and multi-factor authentication in Microsoft Windows operating systems.

- *Axidian Access RDP Windows Logon license*. Grants the right to use the RDP Windows Logon module for two-factor authentication using one-time passwords during terminal access to Microsoft Windows operating systems.
- *Axidian Access Enterprise Single Sign-On license*. Grants the right to use the Enterprise Single Sign-On module for organizing end-to-end authentication in desktop and web applications on end users' workstations.
- *Axidian Access Identity Provider license*. Grants the right to use the Identity Provider module for building a Web Single Sign-On system using the SAML and OpenID Connect protocols.
- *Axidian Access NPS RADIUS Extension license*. Grants the right to use the RADIUS module for implementing two-factor authentication in applications compatible with the RADIUS protocol (VPN, VDI, and others).
- *Axidian Access IIS Extension license*. Grants the right to use the IIS module for implementing strong and two-factor authentication in web applications based on the IIS (Internet Information Services) server.
- *Axidian Access API license*. Grants the right to use the API for integration with third-party systems.
- *Axidian Access ADFS Extension license*. Grants the right to use the ADFS module for building a system using the ADFS protocol.
- *Axidian Access Mobile Device Provisioning license*. Grants the right to use the mobile device provisioning service for releasing devices from quarantine.

System requirements

Server components

System requirements for Axidian Access server components

Integration modules

System requirements for Axidian Access integration modules

Server components

This section includes system requirements for the Axidian Access server components.

Axidian Core Server

Domain	Active Directory
Operating system	Windows Server 2012 R2 and higher
IIS version and modules	Internet Information Services 7.0 and higher with the following modules: • Static Content • HTTP Redirection • ASP.NET • .NET Extensibility • ISAPI Extensions • ISAPI Filters • Basic Authentication • Windows Authentication • IIS Management Console
Microsoft components	Microsoft .NET Framework 4.5.2
Microsoft SQL Server	Microsoft SQL Server 2012 all editions and newer versions
PostgreSQL	• PostgreSQL • PostgreSQL Pro
Hardware requirements	• At least 4 GB RAM • At least 50 GB free disk space

NOTE

To automatically install all the required IIS components, use the PowerShell script included in the distribution package.

Axidian Log Server

Operating System	Windows Server 2012 R2 and higher
IIS Version and Modules	Internet Information Services 7.0 and higher with the following modules: • Static Content • HTTP Redirection • ASP.NET • .NET Extensibility • ISAPI Extensions • ISAPI Filters • Basic Authentication • Windows Authentication • IIS Management Console
Microsoft components	Microsoft .NET Framework 4.6.1
Microsoft SQL Server	Microsoft SQL Server 2012 all editions and newer versions (when using data storage in SQL)
PostgreSQL	PostgreSQL and PostgreSQL Pro 10 and higher
Hardware requirements	• At least 4 GB RAM • At least 50 GB free disk space

> **NOTE**
> To automatically install all the required IIS components, use the PowerShell script included in the distribution package.

Axidian Management Console

> **NOTE**
> If you plan to use Windows Authentication to log in to Management Console, your machine must be in the domain.

Operating System	Windows Server 2012 R2 and higher
IIS Version and Modules	Internet Information Services 7.0 and higher with the following modules: • Static Content • HTTP Redirection • ASP.NET • .NET Extensibility • ISAPI Extensions • ISAPI Filters • Basic Authentication • Windows Authentication • IIS Management Console
Microsoft components	Microsoft .NET Framework 4.8 and higher
Latest version of web browser	• Google Chrome • Mozilla Firefox • Microsoft Edge • Internet Explorer
Hardware requirements	• At least 4 GB RAM • At least 50 GB free disk space

> **NOTE**
> To automatically install all the required IIS components, use the PowerShell script included in the distribution package.

Axidian User Console

> **NOTE**
> If you plan to use Windows Authentication to log in to User Console, the machine must be in the domain.

Operating System	Windows Server 2012 R2 and higher

IIS version and modules	Internet Information Services 7.0 and higher with the following modules: • Static Content • HTTP Redirection • ASP.NET • .NET Extensibility • ISAPI Extensions • ISAPI Filters • Basic Authentication • Windows Authentication • IIS Management Console
Microsoft components	Microsoft .NET Framework 4.5.2
Latest version of web browser	• Google Chrome • Mozilla Firefox • Microsoft Edge • Internet Explorer
Hardware requirements	• At least 4 GB RAM • At least 50 GB free disk space

NOTE

To automatically install all the required IIS components, use the PowerShell script included in the distribution package.

Axidian Phone Management Server

Operating system	Windows Server 2012 R2 and higher

IIS version and modules	Internet Information Services 7.0 and higher with the following modules: • Static Content • HTTP Redirection • ASP.NET • .NET Extensibility • ISAPI Extensions • ISAPI Filters • Basic Authentication • IIS Management Console • URL Authorization — the installation of this component is not included in the Powershell script for automatic installation of IIS components.
Microsoft components	• Microsoft .NET Framework 4.5.2 • Web Deploy Package
Hardware requirements	• At least 4 GB of RAM • At least 50 GB of free disk space

> **NOTE**
>
> For automatic installation of all the required IIS components, use the PowerShell script included in the distribution package.

Axidian Key Server

> **NOTE**
>
> The server supports operation outside the domain.

Operating System	Windows Server 2012 R2 and higher

IIS version and modules	Internet Information Services 7.0 and higher with the following modules: • Static Content • HTTP Redirection • ASP.NET 4.6 • .NET Extensibility • ISAPI Extensions • ISAPI Filters • Anonymous Authentication • IIS Management Console
Microsoft components	Microsoft .NET Framework 4.6.2
Microsoft SQL Server	• Microsoft SQL Server 2012 SP2 all editions • Microsoft SQL Server 2014 all editions • Microsoft SQL Server 2016 all editions • Microsoft SQL Server 2019 all editions
PostgreSQL	• PostgreSQL • PostgreSQL Pro
Hardware Requirements	Hardware requirements match the requirements for the operating systems on which the software operates. • At least 8 GB of RAM • At least 200 GB of free disk space

Interaction with the Axidian Key mobile application	• To send push notifications, the Axidian Key server requires external internet access (to access *fcm.googleapis.com:443* and *oauth2.googleapis.com:443*).
	• The Axidian Key server must be accessible via the DNS name and port specified in the settings (these parameters are set during Axidian Key configuration) from the external network.
	• The DNS server name of Axidian Key must not contain an underscore _.
	• To send push notifications to Huawei devices, access to the HUAWEI Push service is required at the following addresses:
	  ◦ push-api.cloud.huawei.com
	  ◦ oauth-login.cloud.huawei.com

Certificate requirements for iOS devices

> **NOTE**
> You must install the certificate on the device and enable trust to the certificate.

> **INFORMATION**
> The extendedKeyUsage (EKU) extension and expiration date requirements appeared in the fall of 2019 for certificates issued after July 1, 2019. Certificates issued earlier are accepted by iOS-based devices without specifying the object ID and expiration date.

- Certificates and their issuing certificate authorities using RSA keys must use keys of 2048 bits or more.
- Certificates and their issuing certification authorities must use a hashing algorithm from the SHA-2 family to create a digital signature.
- Certificates must contain the name of the DNS server using the Subject Alternative Name extension of the certificate.
- Certificates must include the extendedKeyUsage (EKU) extension containing the id-kp-serverAuth object identifier.
- The validity period of the server certificates should be 825 days or less.

Integration modules

Software requirements

To install each module, you need a deployed Axidian Access system with registered licenses for the corresponding module.

How licensing works in Axidian Access

Axidian ADFS Extension

- Windows Server 2012 R2 and higher;
- The *Active Directory Federation Services* role installed;

Axidian NPS RADIUS Extension

- Windows Server 2012 R2 and higher;
- The *Network Policy Server* role installed;

Axidian IIS Extension

- Windows Server 2012 R2 and higher;
- An installed Internet Information Services server;

Axidian RDP Windows Logon

- Windows Server 2012 R2 and higher;
- Windows 7 32/64bit;
- Windows 8.1 32/64bit;
- Windows 10 32/64bit;
- Windows 11 32/64bit.

Axidian Identity Provider

> **NOTE**
>
> To automatically install all the required IIS server components, use the PowerShell script included in the distribution.

Operating system	Windows Server 2012 R2 and higher

IIS version and modules	Internet Information Services 7.5 and higher with the following modules: • Static Content • Static Content • Default Document • Directory Browsing • HTTP Errors • HTTP Redirection • WebDAV Publishing • ASP.NET • .NET Extensibility • ASP • CGI • ISAPI Extensions • ISAPI Filters • Server Side Includes (SSI) • HTTP Logging • Logging Tools • Request Monitor • Tracing • Basic Authentication • Windows Authentication • Digest Authentication • Client Certificate Mapping Authentication • IIS Client Certificate Mapping Authentication • URL Authorization o Request Filtering • IP and Domain Restrictions • Static Content Compression • Dynamic Content Compression • IIS Management Console • IIS Management Scripts and Tools • Management Service
Microsoft components	.NET 6.0

Components for activation on Windows Server 2012 R2	• HTTP Activation • Message Queuing Activation • TCP Activation • TCP Port Sharing
Client-side environment requirements	• Internet Explorer 11 • Google Chrome 42.0 and higher • Mozilla FireFox 37.0 and higher • Opera 28.0 and higher • MS Edge

Axidian Mobile Device Provisioning

Operating system	Windows Server 2012 R2 and higher
Microsoft components	.NET 6.0
IIS version and modules	Internet Information Services 7.0 and higher with the following modules: • Static Content • HTTP Redirection • ASP.NET • .NET Extensibility • ISAPI Extensions • ISAPI Filters • Basic Authentication • Windows Authentication • IIS Management Console
Client-side environment requirements	• Internet Explorer 11 • Google Chrome 42.0 and higher • Mozilla FireFox 37.0 and higher • Opera 28.0 and higher • MS Edge

Hardware requirements

- At least 4 GB of RAM;
- At least 50 GB of free disk space.

Axidian Windows Logon

Hardware requirements for user workstations	• at least 50 MB of free disk space • .NET Framework version 4.5 or higher
Supported operating systems	• Windows 7 SP1 32/64bit • Windows 8.1 32/64bit • Windows 10 32/64bit • Windows 11 32/64bit • Windows Server 2012/2012 R2 • Windows Server 2016 • Windows Server 2019 • Windows Server 2022
Environment requirements	A configured DNS server (Reverse lookup zones must be added). The DNS server parameters must be specified in the network connection settings on each workstation.
Required administrative rights	To install Axidian Windows Logon, local Administrator rights are required.

Firewall settings	For the correct operation of the Axidian Access system, change the following firewall settings on the Axidian Access Server and on the workstations where the Axidian Windows Logon component is installed: • Open port 53 (DNS) (TCP and UDP) for all processes in both directions. This port is used by the Axidian Access system to determine network availability. • Open port 135 (RPC) for all processes in both directions. This port is used by the system for communication between user workstations and Core Server. • Open port 389 (LDAP) for all processes in both directions. This port is used by the system to gain access to Active Directory (including when searching for Core Server).

Axidian Enterprise Single Sign-On

Hardware requirements for user workstations	• at least 50 MB of free disk space
	• .NET Framework version 4.5 or higher
	• **Web browsers:**
	  ◦ Internet Explorer 8 and higher
	  ◦ Mozilla Firefox 52.6 and higher
	  ◦ Google Chrome 49.0 and higher
	• USB port for connecting an authentication device

Supported operating systems	• Windows Server 2012/2012 R2 • Windows Server 2016 • Windows Server 2019 • Windows Server 2022 • Windows 7 SP1 32/64bit • Windows 8.1 32/64bit • Windows 10 32/64bit • Windows 11 32/64bit

Firewall settings	Set the following firewall settings on the Axidian ESSO server and on the workstations with the installed Axidian Access ESSO Agent: • Open port 53 (DNS) (TCP and UDP) for all processes in both directions. This port is used by the Axidian Access system to determine network availability. • Open port 135 (RPC) for all processes in both directions. This port is used by the system for communication between user workstations and Axidian Access Enterprise Server. • Open port 389 (LDAP) for all processes in both directions. This port is used by the system to gain access to Active Directory (including when searching for Axidian Access Enterprise Server).

Pre-installation steps

Components layout

Axidian Access components requirements

Network

Axidian Access components interaction

Internet Information Services components

Axidian Access required components

Data storage

Contains data for Core Server, Log Server, and Key Server

User catalog

Contains data about Axidian Access users

Client certificate

Verifies requests to the server

Components layout

Layout #1

This layout is recommended for pilot implementations to familiarize yourself with Axidian and perform demonstrations. All system components, required providers, and integration modules are installed on one computer. Integration modules can be deployed on different computers.

Server name	Requirement	Axidian Access components
Core Server PC	Required component	Core Server, Log Server, Management Console
Core Server PC	Optional component	Identity Provider
Core Server PC	Required component. The type of authentication method depends on the final scenario.	Axidian Access providers
Core Server PC	Required component. The type of integration module depends on the final scenario.	Axidian Access integration modules
Axidian Access External PC	Optional component	Axidian Key Server, User Console

System requirements

Server name	CPU	RAM	HDD	Network
Core Server PC	4 cores	8 GB	200 GB	100 Mbps
Axidian Access External PC	4 cores	8 GB	100 GB	100 Mbps

Layout #2

This layout is recommended for production implementations. Critical system nodes are duplicated to ensure fault tolerance. The Axidian Access Management Console component is installed on one of

the Axidian Access servers (Core Server). To configure load distribution between components, you must use a third-party load balancer.

Server name	Requirement	Axidian Access components
Core Server\Log Server	Required component	Core Server, Log Server
Core Server\Management Console\Identity Provider	Required component. Identity Provider is required only for scenarios where authenticator enrollment for external users is needed.	Core Server, Management Console, Identity Provider
Axidian Key Server\User Console	Optional component. Axidian Key Server is required only for scenarios using push authentication. User Console is required for authenticator enrollment for external system users.	Axidian Key Server, User Console
Axidian Key Server\User Console	Optional component. Only for scenarios using push authentication.	Axidian Key Server, User Console
Axidian Access Extensions	Required component. The type of integration module depends on the final scenario.	Axidian Access integration modules
Load balancer	System requirements depend on the selected load balancer.	Load balancing is performed by third-party components.

System requirements

The table shows the recommended system requirements for deploying a production infrastructure for up to 10,000 users.

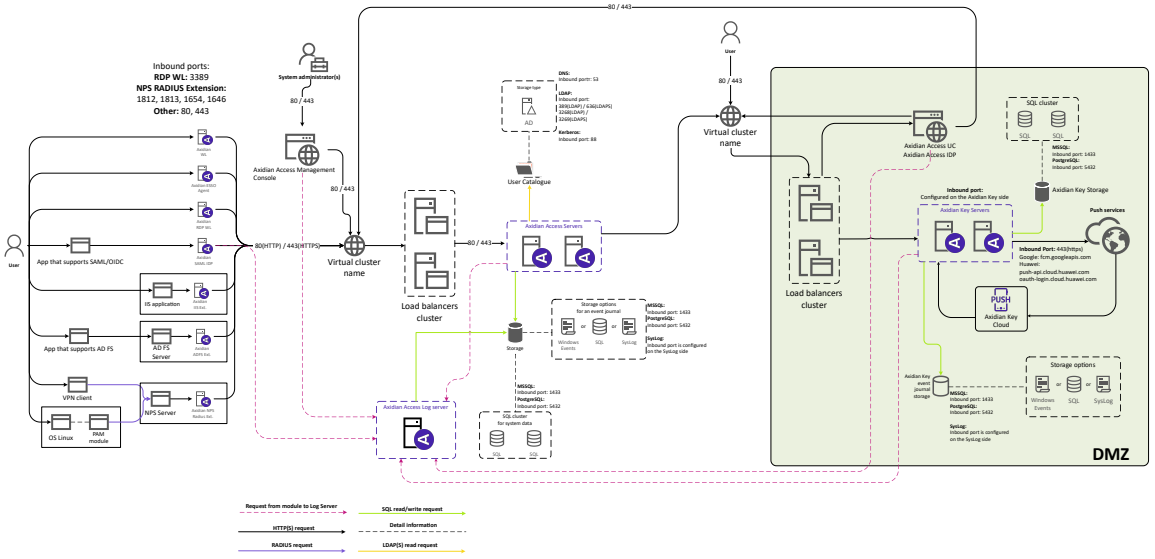
Server name	CPU	RAM	HDD	Network
Core Server\Log Server	8 cores	8 GB	200GB	100 Mbps
Core Server\Management Console\Identity Provider	8 cores	8 GB	200GB	100 Mbps

Server name	CPU	RAM	HDD	Network
Axidian Key Server\User Console	8 cores	8 GB	150GB	100 Mbps
Axidian Key Server\User Console	8 cores	8 GB	150GB	100 Mbps
Axidian Access Extensions	System requirements depend on the selected integration module			
PC with Axidian Access DB and Axidian Access events #1	8 cores	8 GB	500GB	100 Mbps
PC with Axidian Access DB and Axidian Access events #2	8 cores	8 GB	500GB	100 Mbps

Network

The diagram provides a comprehensive overview of Axidian Access component interactions. You can adapt it based on your use cases and requirements.

Load balancers represent optional system elements, they are not built into Axidian Access and can be used in any quantity. The number of the Core Server, Axidian Key Server, Log Server components can also vary depending on client needs.



Interaction between main Axidian Access components in the network infrastructure is carried out through HTTP and HTTPS protocols.

Name	Protocol	Incoming port	Outgoing port
Core Server	TCP/UDP	80(http) 443(https)	49152- 65535
Management Console	TCP/UDP	80(http) 443(https)	49152- 65535

Name	Protocol	Incoming port	Outgoing port
Log Server (Windows Event)	TCP/UDP	80(http) 443(https)	49152- 65535
Log Server (MS SQL)	TCP/UDP	1433	49152- 65535
Log Server (PostgreSQL)	TCP/UDP	5432	49152- 65535
Axidian Access MS SQL Database	TCP/UDP	1433	49152- 65535
Axidian Access PostgreSQL Database	TCP/UDP	5432	49152- 65535
Active Directory	DNS	53	49152- 65535
	LDAP	389 (LDAP) / 636 (LDAPS)3268 (LDAP) / 3269 (LDAPS)	49152- 65535
	Kerberos	88	49152- 65535
Axidian Key Server	TCP/UDP	Configured on the AK server side	49152- 65535
Axidian Key Server MS SQL Database	TCP/UDP	1433	49152- 65535
User Console	TCP/UDP	80(http) 443(https)	49152- 65535
ADFS Extension	TCP/UDP	80(http) 443(https)	49152- 65535
IIS Extension	TCP/UDP	80(http) 443(https)	49152- 65535
NPS RADIUS Extension	TCP/UDP	80(http) 443(https)	49152- 65535
RDP Windows Logon	TCP	3389	49152- 65535
	TCP/UDP	80(http) 443(https)	49152- 65535
Identity Provider	TCP/UDP	80(http) 443(https)	49152- 65535

Internet Information Services components

Before you install any Axidian Access module, you must install the required Internet Information Services components.

To install Internet Information Services components, use a Powershell script that you can find in the Axidian Access distribution package. Otherwise, install the components manually.

You can find the list of required Internet Information Services components in the system requirements for server components and integration modules.

> ⚠ **IMPORTANT**
> Run PowerShell as administrator.

To locate the script, open the folder with the Axidian Access distribution and go to the *Misc\IISScripts* folder:

- For server components: Use the *Axidian.Main.IIS.Install.MSServer.ps1* script.
- For server components and the Identity Provider module: Use the *Axidian.SAML.IIS.Install.MSServer.ps1* script.

If your security policy does not allow running unsigned scripts, run the following command to lift the restriction:

```
Set-ExecutionPolicy Unrestricted
```

If you cannot lift the restriction, open the script in a text editor, copy the entire contents, and paste it into PowerShell — the script will run.

Data storage

For Axidian Access, you must create and configure the following databases:

- For the Core Server component.
- For the Log Server component.
- If you plan to use the Axidian Key mobile application, you need a separate database for the Key Server component.

Also, you must create a service user that can perform requests for reading and writing information.

Axidian Access supports Microsoft SQL and PostgreSQL databases.

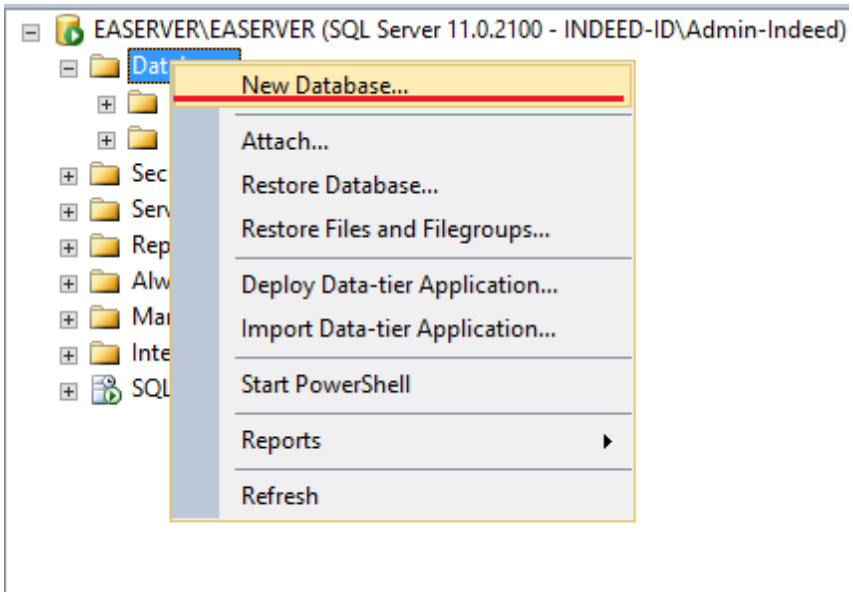
Microsoft SQL

On the server with Microsoft SQL Server installed, create two databases: one for Core Server, the other one for Log Server. Also, create a service account.

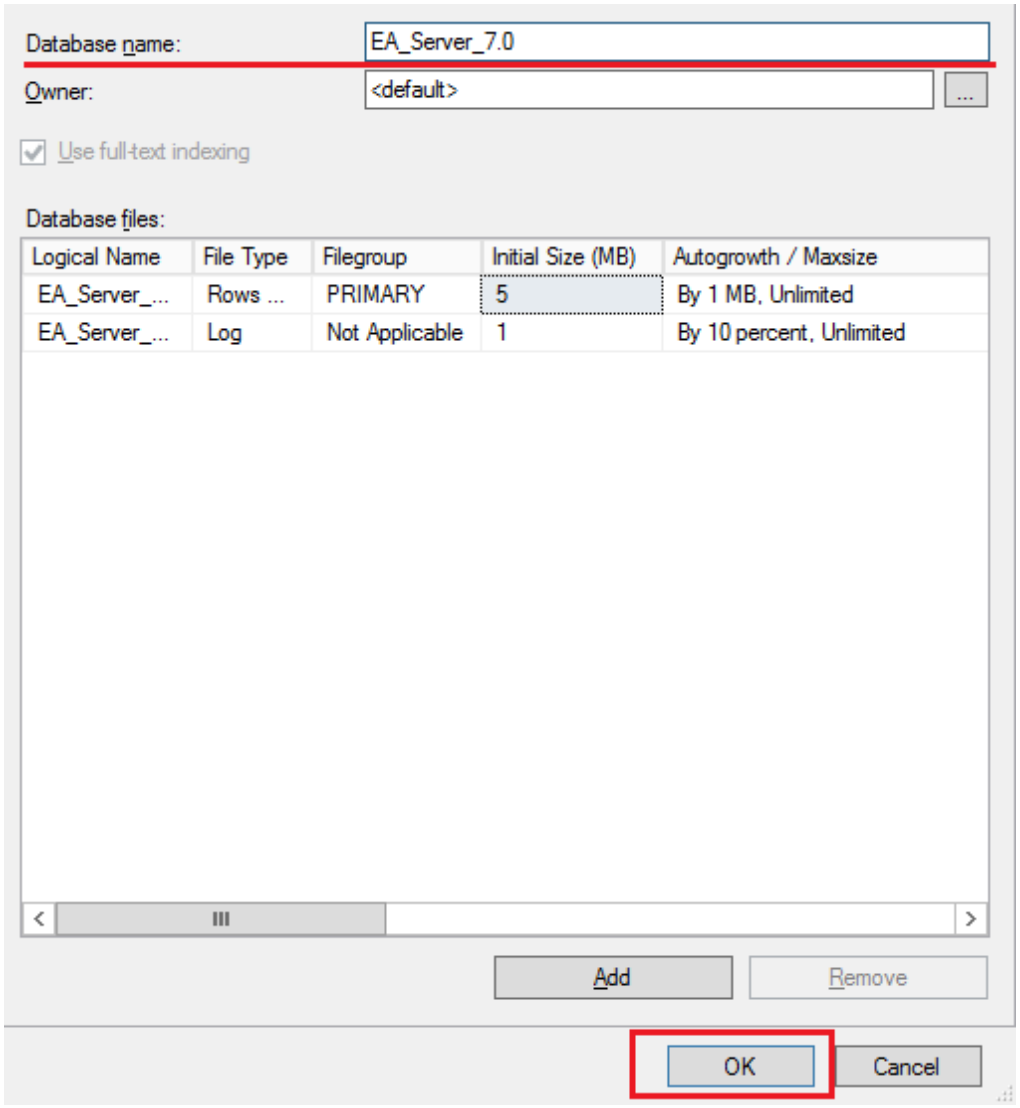
Create a database

To create a database, perform the following actions:

1. Open SQL Server Management Studio and connect to the server.
2. Right-click the **Databases** node.
3. In the window that opens, select **New Database**.



4. Enter the name of the new database and click **OK**.



Create a service account

Axidian Access supports service accounts with Windows authentication mode and with SQL Server authentication mode.

> ⚠ **IMPORTANT**
>
> When creating a service account, disable mandatory password change at next login and password expiration. Otherwise, when the password expires, Axidian Access components will not be able to access the database, which will cause failures.

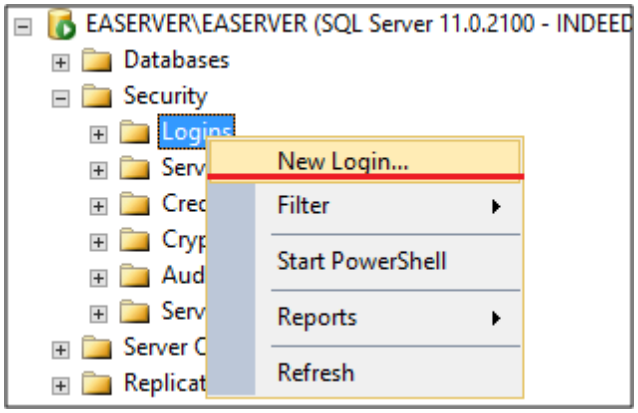
Create a service account with SQL Server authentication mode

To create a service account with SQL Server authentication mode, perform the following actions:

1. Expand the **Security** node.

2. Right-click the **Logins** node.

3. In the window that appears, select **New Login**.



4. In the **Login - New** window, perform the following actions:

    1. Enter the user name.
    2. Select the **SQL Server authentication** check box.
    3. Clear the **Enforce password Policy** check box.
    4. In the left menu, select **User Mapping**.
    5. Select the previously created databases.
    6. For each database, select the **db_owner** role and click **OK**.

5. Enable mixed authentication:

    1. Right-click the SQL Server instance node and select **Properties**.
    2. On the **Security** page in the **Server authentication** section, select **SQL Server and Windows Authentication** mode.

6. Verify that the service account has the following setting enabled:

    1. Right-click the created account and select **Properties**.
    2. On the **Status** page in the **Login** section, select **Enabled**.

7. Open **Sql Server Configuration Manager** and enable the TCP/IP protocol:

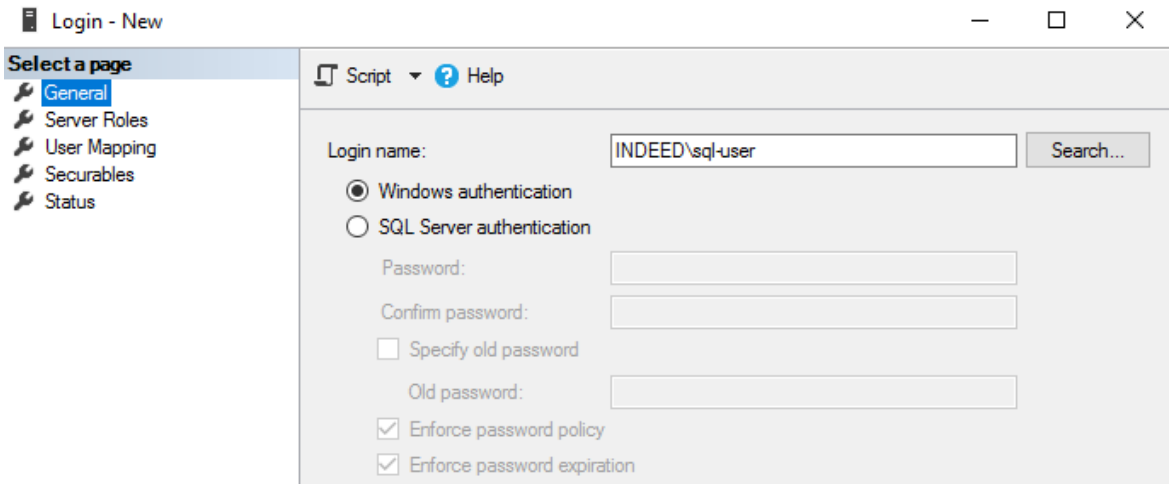
    1. On the **Native client SQL** tab, select **Client protocols**.
    2. Right-click **TCP/IP**, select **Properties** and set the value to **Yes** to enable the TCP/IP protocol.

3. Enable TCP/IP in all **Native client SQL Client Configuration** tabs.

Create a service account with Windows authentication mode

To create a service account with SQL Server authentication mode, perform the following actions:

1. Create a domain account.
2. Open SQL Server Management Studio and connect to the server.
3. Expand the **Security** node.
4. Right-click the **Logins** node.
5. In the window that appears, select **New Login**.
6. In the **Login - New** window, perform the following actions:
    1. Find the previously created user.
    2. In the left menu, select **User Mapping**.
    3. Select the previously created databases.
    4. For each database, select the **db_owner** role and click **OK**.



7. Open Local Group Policy Editor and perform the following actions:
    1. Expand **Computer Configuration→Windows Settings→Security Settings→Local Policies→User Rights Assignment**.
    2. In the list on the right, select the **Log on as a service** policy.

3. Click **Add User or Group**, enter the name of the previously created user, and click **Check Names**.
4. Click **OK** to add the user to the policy.

8. In a text editor, open the configuration file of the component for which you are creating the database:
   1. The file for Core Server is located in *C:\inetpub\wwwroot\am\core\Web.config*.
   2. The file for Log Server component is located in *C:\inetpub\wwwroot\ls\targetConfigs\sampleDb.config*.
9. In the `connectionString` tag, specify the value `Integrated Security=SSPI`.

Example for Core Server

```
<mssqlDbContexts>
      <mssqlDbContext connectionString="Data Source=SERVER\SQLEXPRESS;Initial
Catalog=am8;Integrated Security=SSPI" id="mssql" />
</mssqlDbContexts>
```

Example for Log Server

```
<Settings>
  <ConnectionString>Data Source=SERVER\SQLEXPRESS;Initial Catalog=Log81;Integrated
Security=SSPI</ConnectionString>
</Settings>
```

10. Save the configuration file.

Next, configure the application pools in Internet Information Services.

1. Start IIS Manager.
2. Click **Application Pools** and select:
   - **Axidian.Core** for Core Server.
   - **Axidian.LS** for Log Server.
3. In the **Actions** section, select **Advanced Settings**.
4. In the **Advanced Settings** window, find the **Identity** parameter and click the button on the right.

5. In the **Application Pool Identity** window, select **Custom account** and click **Set**.
6. Specify the service account with the domain and click **OK**.
7. In the **Actions** section, restart the application pool for which the setting is configured, and open the application start page in IIS.
8. Select the customizable Axidian server component in IIS.

> **INFORMATION**
> If the setting is configured for both Core and Log Server, perform these steps for each component.

9. In the **Management** section, select **Configuration Editor** and open *system.webServer/security/authentication/windowsAuthentication*.
10. For the `useAppPoolCredentials` parameter, set the value to `True` and in the **Actions** block, click **Apply**.
11. Restart IIS.

PostgreSQL

On the server with PostgreSQL Server installed, create two databases: one for Core Server, the other one for Log Server. Also, create a service account.

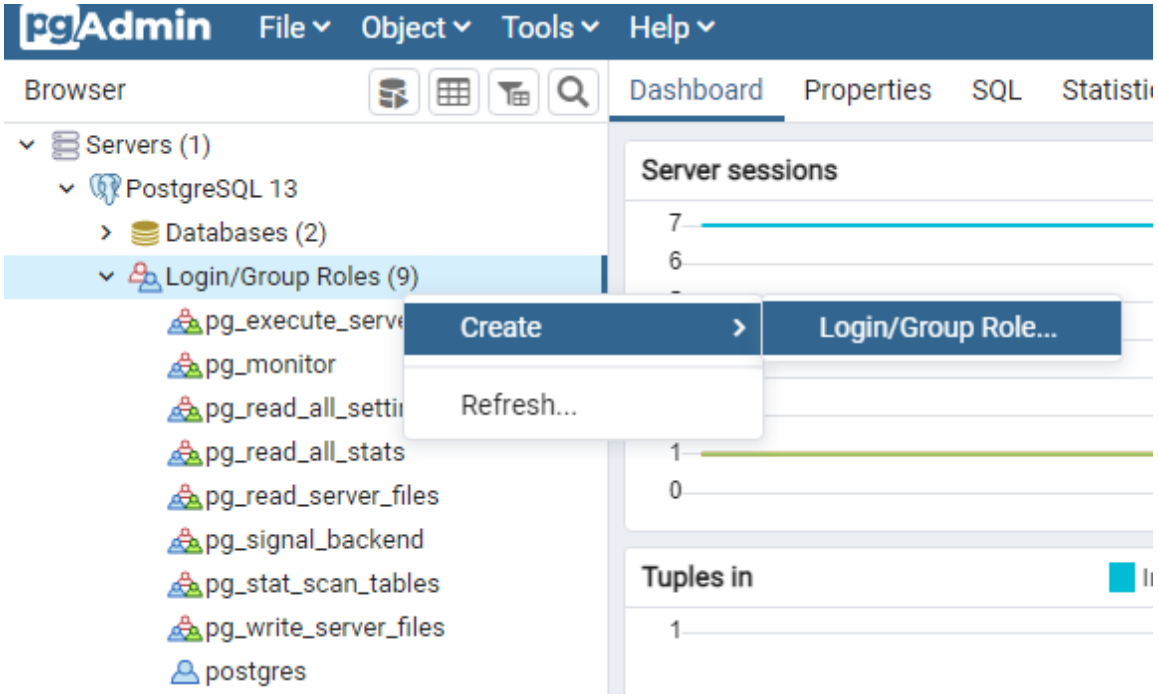
If the databases and Core Server are located on different machines, you must configure remote connection to the database.

Create a service account

To create a service account, perform the following actions:

1. Open *pgAdmin* and connect to the server.

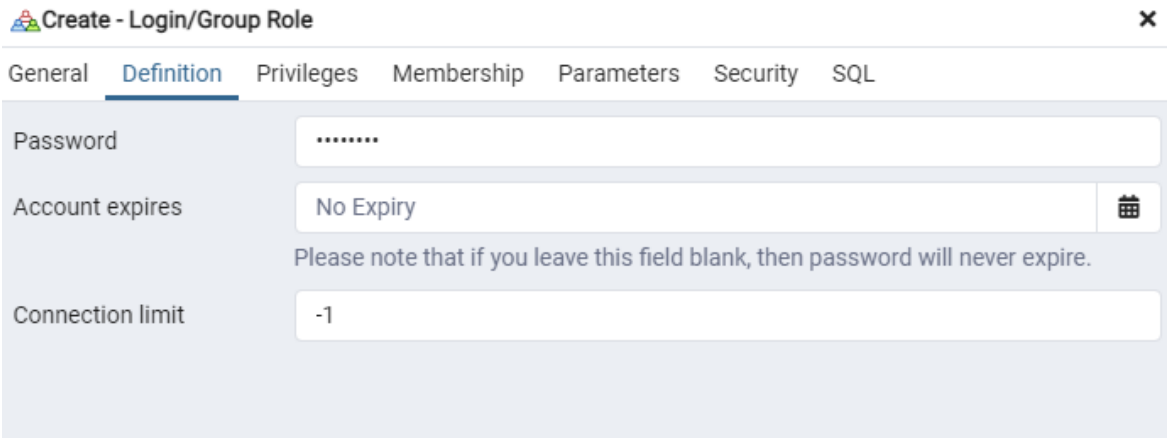
2. Right-click **Login/Group Roles** and select **Create** → **Login/Group Role**.



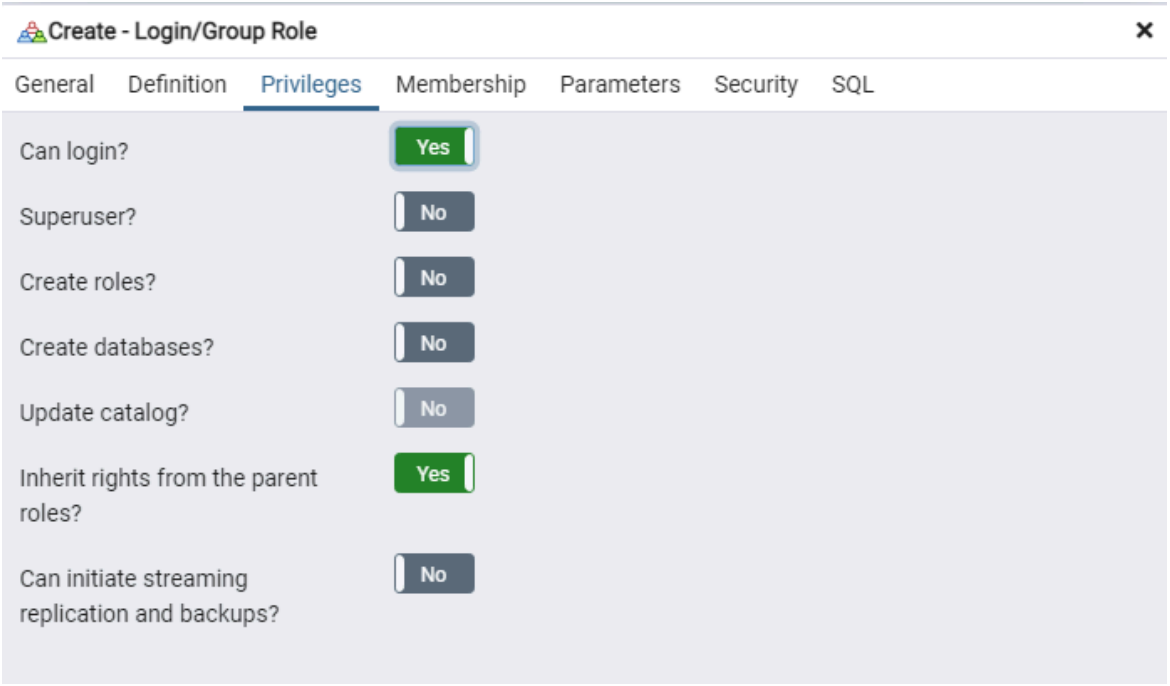
3. In the **Create-Login/Group Role** window, perform the following actions:

    1. On the **General** tab, in the **Name** field, enter the user name.

    2. On the **Definition** tab, in the **Password** field, enter the password. In the **Account Expires** field, specify the value **No Expiry**.



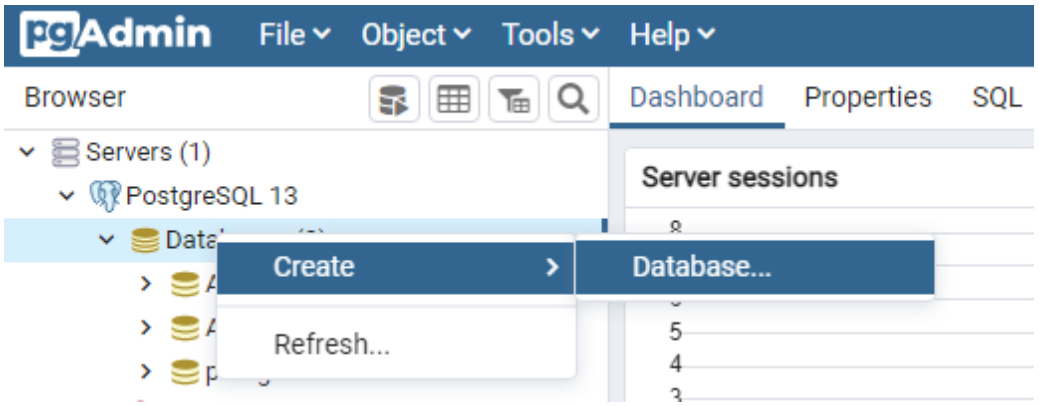
3. On the **Privileges** tab, select the **Can login?** check box and click **Save**.



Create a database

To create a database, perform the following actions:

1. Open *pgAdmin* and connect to the server.
2. Expand the components of your SQL server.
3. Right-click **Databases**.
4. Select **Create** → **Database**.



5. In the **Create-Database** window, perform the following actions:
    1. In the **Database** field, enter the database name.
    2. In the **Owner** field, select from the list the previously created user and click **Save**.

Configure remote connection to the database

To configure remote connection to the database, perform the following actions:

1. Open the configuration file *pg_hba.conf* from *C:\Program Files\PostgreSQL<version number>\data*.

2. Add a string in the following format:

   ```
   host databaseName UserName HostIP md5
   ```

   Where:

   - `host`: Connection type. If you do not specify a connection type, the TCP/IP connection is used by default.
   - `databaseName`: Name of the database for which you are configuring the connection.
   - `UserName`: Name of the user who connects to this database.
   - `HostIP`: IP address of the machine on which the Axidian component is installed. Specify the value with a mask.
   - `md5`: User authentication method.

     Example

     ```
     host all all 192.168.1.0/24 md5
     ```

3. Open the configuration file *postgresql.conf* from *C:\Program Files\PostgreSQL<version number>\data*.

4. Change the string `listen\_addresses = 'localhost'` to `listen\_addresses = '\*'`.

User catalog

Axidian Access uses end user data from Active Directory. These are the users who goes through the authentication procedure using Axidian Access.

For user data to appear in Axidian Access, you must prepare an object with end users in Active Directory and create a service account that can read data.

You can configure a separate user organizational unit or use the entire user domain.

> ⚠ **IMPORTANT**
> If you plan to use a separate organizational unit, all necessary objects (groups, administrators, end users) must be located within this organizational unit.

Axidian Access only reads data from Active Directory, no new data is written.

Create a service account

To create a service account, perform the following action:

1. Open Active Directory Users and Computers.
2. Select the catalog where you plan to create the service account.
3. Click the icon  .
4. In the window that appears (**New Object** → **User**), enter the user name and click **Next**.
5. Enter the password.
6. Clear **User must change password at next logon**.
7. Select **Password never expires** and click **Finish**.

Grant data read permissions

If your company security policy allows all users in the domain to read data of all objects in the domain, skip the following steps.

If a data read restriction is configured, perform the following action:

1. Open Active Directory Users and Computers.

2. On the **View** tab, select the **Advanced Features** check box.

3. Right-click the object with users and select **Properties**.

4. In the **Properties** window, perform the following action:

    1. Open the **Security** tab.
    2. In the **Group or user names** section, add the previously created service account.
    3. In the **Permissions for** section, select the **Read** option and click **Advanced**.

5. In the **Advanced Security Settings** window, select the service account and click **Edit**.

6. In the **Permission Entry** window, from the **Applies to** list, select **This object and all descendant objects** and click **OK**.

Active Directory user attributes

User attribute	Settings
objectGUID	Global unique object identifier
objectSid	A unique security identifier assigned to each object in Active Directory. Used for managing access to resources and authentication of users and objects
distinguishedName	A unique name of an object in the Active Directory hierarchy consisting of the full path to the object. For example: *CN=testuser,OU=Users,DC=domain,DC=com*
tokenGroups	A computed attribute containing a list of SIDs of all groups, including nested ones, which the user belongs to when logging in to the system
memberOf	Attribute containing the distinguishedName of groups of which the user or computer is a direct member
sAMAccountName	Account name of the user, group, or computer used to support compatibility with previous versions of Windows. For example, *ivanov*
msDS-PrincipalName	Canonical name of the user. For example: *DOMAIN\username*

Client certificate

The client certificate verifies requests from integration modules to Core Server.

Certificate installation is required only for direct requests from modules to the server. If you use a load balancer, a certificate is optional. However, it is required when requests are not sent through the load balancer.

Certificate *SourceCode_EA.Server_Certificates_ClientCertificate_eaclient* is located at *Axidian AM version_number\Axidian Access Server\version_number\Misc*.

Install the certificate from the Axidian Access installation package on the computer with the deployed Core Server.

To do this, perform the following actions:

1. Double-click the certificate file.
2. In the **Certificate** window, click **Install Certificate**.
3. In the **Certificate Import Wizard** window, select **Local Computer** and click **Next**.
4. Allow making changes.
5. In the **Certificate Import Wizard** window, select **Place all certificates in the following store** and click **Browse**.
6. In the **Select Certificate Store** window, select **Trusted Root Certification Authorities** and click **OK**.
7. In the **Certificate Import Wizard** window, click **Next** → **Done**.

To verify that you have allowed the Axidian Access server to accept the client certificate, follow these steps:

1. Open IIS Manager.
2. Navigate to **Axidian Access** → **Sites** → **Default Web Site** → **am** → **core**.
3. Click **SSL Settings**.
4. Select **Accept client certificates**.

Install server components

Log Server

Server for managing Axidian Access log

Core Server

5 items

Management Console

Web interface to administer Axidian Access

User Console

Web interface for Axidian Access users

Key Server

The server for the operation of Axidian Key

Reset Password

A utility for resetting the user's password

Location of configuration files

How to find configuration files

Log Server

Axidian Log Server (Log Server) is a web application that runs on IIS. This module is responsible for the centralized collection and audit of system events.

Events are recorded in databases.

How do I create and configure a database?

To configure Log Server:

1. Install Log Server and EventLog.
2. Create an HTTPS binding in the IIS settings.
3. Configure Log Server according to the data storage you use.
4. If required, change the event caching settings.

Installation files

The files for the Log Server are located in the *Axidian Access LogServer* folder.

- *Axidian.LogServer-<version number>.x64.en-us.msi*: Log Server installation package.
- *Axidian.Server.EventLog-<version number>.x64.en-us.msi*: Package for creating an Axidian Access event schema and the required log structure in Windows EventLog.
- *Axidian.LogServer.Config.Encryptor.zip*: Utility for encrypting a configuration file.

Install Log Server and EventLog

System requirements

1. To install Log Server, start the corresponding package.

2. To install EventLog on a computer with Log Server, run the *Axidian.Server installer.EventLog-version numbers.x64.en-us.msi*.

The event schema is created automatically when installing the Axidian.Server component.EventLog and it is located at *C:\ProgramData\Axidian\LogServer\Schemes*.

HTTPS binding in IIS Manager

Log Server is a web application that runs on the AIS database. During the installation process, SSL is required in the settings for Log Server, which in turn requires HTTPS connectivity.

1. Start IIS Manager and expand the **Sites** item.
2. Select **Default Web Site** and in the **Actions** section, click **Bindings**.
3. Click **Add** and configure the following options:
    - Type: https.
    - Port: 443.
    - Select SSL Certificate.
4. Save the link.

If you do not plan to use the HTTPS protocol, disable the SSL requirement in the IIS settings for Log Server.

Configure Log Server with different types of data storage

Select and configure the supported data storage:

- Microsoft SQL or PostgreSQL databases
- EventLog storage
- Storage in SysLog
- Backup storage

Microsoft SQL

Microsoft SQL

1. Open *C:\inetpub\wwwroot\ls\targetConfigs\sampleDb.config* for editing.
2. Specify the data to connect to the database in the `ConnectionString` tag.
   - `Data Source`: Server instance. This property is mandatory for all compounds. Acceptable values are the network name or IP address of the server, local or localhost for local connections.
   - `Database`: Name of the database.
   - `User Id`: Name of the user to connect to the database.
   - `Password`: User password for connecting to the database.

```
<Settings>
<ConnectionString>Data Source=localhost;Database=LogServ;User
Id=log;Password=Q1q2E3e4</ConnectionString>
</Settings>
```

3. Open *C:\inetpub\wwwroot\ls\clientApps.config*.
4. Locate and uncomment the block with `"applicationId"="ea"` in the `TargetID` and `ReadTargetId` tags to specify sampleDb.

> **INFORMATION**
>
> The `ReadTargetId` tags specify the identifier of the repository where events are read.
>
> In the `WriteTargets` block, the `TargetID` tags specify the identifier of the repository where events are recorded.
>
> The IDs are set in the `Targets` tag, and the configuration files for each type are located in the *targetConfigs* folder with the corresponding name.

```
      <Application Id="ea" SchemaId="eaSchema">
          <ReadTargetId>sampleDb</ReadTargetId>
          <WriteTargets>
            <TargetId>sampleDb</TargetId>
          </WriteTargets>
          <AccessControl>
            <!--<CertificateAccessControl CertificateThumbprint="001122...AA11"
Rights="Read" />-->
          </AccessControl>
        </Application>
```

Encryption/Decryption of the configuration file

1. Run the command prompt as an administrator.
2. At the command prompt, navigate to the folder with the encryption utility *Axidian.LogServer.Config.Encryptor*.

Encrypting the configuration file

For encryption, run the utility with the following parameters:

```
protect -f C:\inetpub\wwwroot\ls\targetConfigs\sampleDb.config -t Sql

 -f, --file Path to the configuration file, required parameter.
  -t, --type is the type of the configuration file target. An optional parameter, which is
Sql by default.
```

Decryption of the configuration file

To decrypt it, run the utility with the following parameters:

```
unprotect -f C:\inetpub\wwwroot\ls\targetConfigs\sampleDb.config -t Sql

 -f, --file Path to the configuration file, required parameter.
  -t, --type is the type of the configuration file target. An optional parameter, which is
Sql by default.
```

PostgreSQL

PostgreSQL

1. Navigate to *C:\inetpub\wwwroot\ls\targetConfigs*.
2. Create a file with an arbitrary name and the *.config* extension, such as *postgresDb.config*.
3. Add the following lines to the file:

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
<ConnectionString>server=server_dns_name;port=5432;user
id=account_name;password=account_password;database=database_name</ConnectionString>
</Settings>
```

4. In the `ConnectionString` line, set the values of the following parameters:
   - `server`: DNS\IP address of the server with the PostgreSQL database.
   - `port`: Port where the connection takes place. The default value is 5432.
   - `user ID`: Name of the user who has full rights to the Database.
   - `password`: User password.
   - `database`: Name of the Log Server database.

   ```
   server=192.200.1.2;port=5432;user id=AxidianDB;password=Passw0rd;database=LogServerDB
   ```

5. Open *C:\inetpub\wwwroot\ls\clientApps.config* for editing.
6. Uncomment the block with `applicationId="ea"` in the `TargetID` and `ReadTargetId` tags, specify `postgresDb`.

   > **INFORMATION**
   >
   > In the `ReadTargetId` tags, the storage identifier is specified, and events are read.
   >
   > The `WriteTargets` block, in the `TargetID` tags, specifies the identifier of the repository where events are recorded.
   >
   > The identifiers are specified in the `Targets` tag, the configuration files for each type are located in the *targetConfigs* folder with the corresponding name.

7. Then, in the same file, in the `Targets` section, add a new element:

```
<Targets>
...
<Target Id="postgresDb" Type="pgsql"/>
</Targets>
```

Display example

Encryption/Decryption of the configuration file

1. Run the command prompt as an administrator.
2. At the command prompt, navigate to the folder with the encryption utility *Axidian.LogServer.Config.Encryptor*.

Encrypting the configuration file

To encrypt, run the utility with the following parameters:

```
protect -f C:\inetpub\wwwroot\ls\targetConfigs\postgresDb.config -t Sql

 -f, --file Path to the configuration file, required parameter.
  -t, --type is the type of the configuration file target. An optional parameter, which is
Sql by default.
```

Decryption of the configuration file

To decrypt, run the utility with the following parameters:

```
unprotect -f C:\inetpub\wwwroot\ls\targetConfigs\postgresDb.config -t Sql

 -f, --file Path to the configuration file, required parameter.
  -t, --type is the type of the configuration file target. An optional parameter, which is
Sql by default.
```

EventLog storage

EventLog storage

Editing the configuration file

1. Open *C:\inetpub\wwwroot\ls\clientApps.config* for editing.
2. Uncommment the block with `"Application Id"="ea"` in the `TargetID` and `ReadTargetId` tags to specify `sampleEventLog`.

> **INFORMATION**
>
> The `ReadTargetId` tags specify the identifier of the repository where events are read.
>
> In the `WriteTargets` block, the `TargetID` tags specify the identifier of the repository where events are recorded.
>
> The IDs are set in the `Targets` tag, and the configuration files for each type are located in the *targetConfigs* folder with the corresponding name.

```
    <Application Id="ea" SchemaId="eaSchema">
      <ReadTargetId>sampleEventLog</ReadTargetId>
      <WriteTargets>
        <TargetId>sampleEventLog</TargetId>
      </WriteTargets>
      <AccessControl>
        <!--<CertificateAccessControl CertificateThumbprint="001122...AA11"
Rights="Read" />-->
      </AccessControl>
    </Application>
```

Display example

Log display in Windows events

Storage in SysLog

Storage in SysLog

The Log Server supports the syslog format, you can use any server that works with this format. As an example, the configuration of the syslog server Syslog Watcher v4.8.6 is further considered.

You can download the utility on the official website

1. Run the installation file *SyslogWatcherSetup-win32.msi*.
2. In the **License Agreement** window, accept the license agreement.
3. In the **Installation Type** window, select the install type **InstallSyslogWatcher Service** and GUI.
4. In the **Select Installation Folder** window, select the installation path for the syslog server.
5. In the **Windows Firewall Exception** window, allow the addition of rules for all incoming connections for Syslog Watcher to Windows Firewall.
6. In the **Confirm Installation** window, click **Next** to confirm the installation.
7. Wait for the server installation to complete.

Configure the sysLog server

1. Launch Syslog Watcher and select **Manage Local Syslog Server**.
2. Click **Settings** in the top menu of the program.
    1. Select **Network Interfaces**.
    2. Verify that the udp protocol is selected and the port is specified.
    3. Select **Processing**, then select the UTF-8 encoding.

Edit the configuration file

1. Open *C:\inetpub\wwwroot\ls\targetConfigs\sampleSyslog.config* and specify the following parameters in the `ConnectionString` tag:
    - *HostName*: Name or IP address of the Syslog server.
    - *Port*: Port of the Syslog server (514 is the default port).
    - *Protocol*: Type of connection to the Syslog server: UDP, TCP, TCPoverTLS.
    - *Format*: Optional parameter that defines the format of logs: Plain (default), CEF, LEEF.

- *SyslogVersion*: Optional parameter, protocol specification: RFC3164, RFC5424.

```
<Settings HostName="localhost" Port="514" Protocol="UDP" Format="Plain"  />
```

2. Open *C:\inetpub\wwwroot\ls\clientApps.config*.

3. For a block with `"Application Id"="ea"`, specify `sampleSyslog` in the `TargetID` tag for `WriteTargets`. Leave *the `ReadTargetId` tag empty, since reading is performed by a third-party program, in this example Syslog Watcher.

> **INFORMATION**
>
> The `ReadTargetId` tags specify the identifier of the repository where events are read.
>
> In the `WriteTargets` block, the `TargetID` tags specify the identifier of the repository where events are recorded.
>
> The IDs are set in the `Targets` tag, and the configuration files for each type are located in the *targetConfigs* folder with the corresponding name.

```
<Application Id="ea" SchemaId="eaSchema">
    <ReadTargetId> </ReadTargetId>
        <WriteTargets>
            <TargetId>sampleSyslog</TargetId>
        </WriteTargets> <AccessControl>
<!--<CertificateAccessControl CertificateThumbprint="001122...AA11" Rights="Read" />-->
</AccessControl>
</Application>
```

4. Click **Start Server** in the upper-left corner of the Syslog Watcher program.

Display example

Backup Storage

Backup storage

Setup with multiple repositories of different types

> **INFORMATION**
> As an example, a setting has been made in which events are read from Windows events and written to Windows events and the SQL database.

1. Open the *C:\inetpub\wwwroot\ls\clientApps.*config*.
2. For a block with `"Application Id"="ea"`, specify `sampleEventLog` or `sampleDb` in the `ReadTargetId` tags.

   > **INFORMATION**
   > The `ReadTargetId` tags specify the identifier of the repository from which events are read.
   >
   > In the `WriteTargets` block, the `TargetID` tags specify the identifier of the repository where events are recorded.
   >
   > The identifiers are specified in the `Targets` tag, and the configuration files for each type are located in the *targetConfigs* folder with the corresponding name.

3. In the `WriteTargets` block, specify `<TargetID>sampleEventLog</TargetID>` and `<TargetID>sampleDb</TargetID>`.
4. Open *C:\inetpub\wwwroot\ls\targetConfigs\sampleDb.config*.
5. Specify the data to connect to the database in the `ConnectionString` tag.
   - `Data Source`: Server instance. This property is mandatory for all compounds. Acceptable values are the network name or IP address of the server, local or localhost for local connections.
   - `Database`: Name of the database.
   - `User Id`: Name of the user to connect to the database.
   - `Password`: User password for connecting to the database.

```
<Settings>
<ConnectionString>Data Source=localhost;Database=LogServ;User
Id=log;Password=Q1q2E3e4</ConnectionString>
</Settings>
```

> **INFORMATION**
>
> With this configuration, system events will be read from Windows events, and events will be written to the SQL database.

```
<Application Id="ea" SchemaId="eaSchema">
<ReadTargetId>sampleEventLog</ReadTargetId>
<WriteTargets>
    <TargetId>sampleEventLog</TargetId>
    <TargetId>sampleDb</TargetId>
</WriteTargets>
<AccessControl>
    <!--<CertificateAccessControl CertificateThumbprint="001122...AA11" Rights="Read"
/>-->
</AccessControl>
</Application>
```

Configuration with backup storage in the database

1. Open *C:\inetpub\wwwroot\ls\clientApps.config*.
2. In the `Targets` block, copy the `Target` tag with `Id="sampleDb"` and change the id to an arbitrary value, such as `sampleDbBackup`.

```
<Targets>
    <Target Id="sampleEventLog" Type="eventlog"/>
    <Target Id="sampleDb" Type="mssql"/>
    <Target Id="sampleDbBackup" Type="mssql"/>
    <Target Id="akcSqlTarget" Type="mssql"/>
    <Target Id="akcEventLogTarget" Type="eventlog"/>
    <Target Id="sampleSyslog" Type="syslog"/>
</Targets>
```

3. For a block with `Application Id="ea"`, specify `sampleDb` in the `ReadTargetId` tags.

> **INFORMATION**
>
> The `ReadTargetId` tags specify the identifier of the repository from which events are read.
>
> In the `WriteTargets` block, the `TargetID` tags specify the identifier of the repository where events are recorded.
>
> The IDs are set in the `Targets` tag, and the configuration files for each type are located in the *targetConfigs* folder with the corresponding name.

4. In the `WriteTargets` block, specify the `<TargetID>sampleDb</TargetID>` and `<TargetID>sampleDbBackup</TargetID>`.

> **INFORMATION**
>
> With this configuration, events will be read from the database specified in the configuration file *ls\targetConfigs\sampleDb.config*. The recording will be performed in the databases specified in the files:
>
> - *ls\targetConfigs\sampleDb.config*
> - *ls\targetConfigs\sampleDbBackup.config*

```
<Application Id="ea" SchemaId="eaSchema">
<ReadTargetId>sampleDb</ReadTargetId>
<WriteTargets>
    <TargetId>sampleDb</TargetId>
    <TargetId>sampleDbBackup</TargetId>
</WriteTargets>
<AccessControl>
    <!--<CertificateAccessControl CertificateThumbprint="001122...AA11" Rights="Read"
/>-->
</AccessControl>
</Application>
```

5. Open *C:\inetpub\wwwroot\ls\targetConfigs\sampleDb.config*.

6. Specify the data to connect to the database in the `ConnectionString` tag:

> **INFORMATION**
>
> Specify the data to connect to the main database.

- `Data Source`: Server instance. This property is mandatory for all compounds. Acceptable values are the network name or IP address of the server, local or localhost for local connections.
- `Database`: Name of the database.
- `User Id`: Name of the user to connect to the database.
- `Password`: User password for connecting to the database.

```
<Settings>
<ConnectionString>Data Source=localhost;Database=LogServ;User
Id=log;Password=Q1q2E3e4</ConnectionString>
</Settings>
```

7. Create a *sampleDbBackup* file with the *.config* extension.

> ⚠ **CAUTION**
> The file name must exactly repeat the value of the Id specified in step 2.

8. Specify the data to connect to the database in the `ConnectionString` tag.

> ⓘ **INFORMATION**
> Specify the data to connect to the backup database.

- `Data Source`: Server instance. This property is mandatory for all compounds. Acceptable values are the network name or IP address of the server, local or localhost for local connections.
- `Database`: Name of the database.
- `User Id`: Name of the user to connect to the database.
- `Password`: User password for connecting to the database.

```
 <Settings>
   <ConnectionString>Data Source=localhost;Database=LogServBackup;User
Id=log;Password=Q1q2E3e4</ConnectionString>
 </Settings>
```

Fault tolerance

If Log Server is unavailable, the events are cached in the local folder on the Axidian Access server.

The information about Log Server unavailability is displayed in Windows events on the **Application** tab.

Event code: 500.

Text: *Log server client fluentSchedulerBackgroundLoaderTask. IterationWEBexception: The waiting time for the operation has expired*.

The path to the folder for storing events is set in the *Web.config* configuration file in the `logServerClient` tag of the `eventcachedirectory` parameter (default: `./EventCache`). The time for sending logs after LogServer is resumed is set in the `Logserverclient` tag of the `EventCacheSendingIntervalSec` parameter (default: `600` seconds).

It is not recommended to set the value for the `EventCacheSendingIntervalSec` parameter to less than 30 seconds.

Core Server

Axidian Core Server (Core Server) is the main server module responsible for:

- Centralized storage of authenticators, passwords, and user settings.
- Centralized management and administration.
- Centralized receiving and processing of requests from other modules of the system.
- Coordinating the actions of individual modules and the system in general.

Core Server caches data when retrieving a user, group, container, authorization scenarios, and business logic. Main catalog requests include retrieving a user, group, container by ID. These requests are from cache. All other requests (by name, by phone, email) only update the cache with data from the catalog. The first user request by ID also updates the cache, subsequent ones are taken from cache. Object lifetime in cache is 10 minutes.

Installation files

Files for the Axidian Access Core Server installation are located at *Axidian Access Server/<version number>*:

- *Axidian.Server-<version number>.x64.en-us.msi*: Package for installing Core Server.
- *\Misc\AccessManager.Tools.KeyGen.Console.exe*: Utility for generating a key pair for user token signing.
- *\Misc\AccessControlInitialConfig\EA.Server.AccessControlInitialConfig.exe*: Initial configuration utility.
- *\Misc\AccessControlInitialConfig\EA.Server.AccessControlInitialConfig.exe.config*: Configuration utility settings file.
- *\Misc\EA.Config.Encryptor\EA.Config.Encryptor.exe*: Utility for encrypting the configuration file.
- *\Misc\EA.Config.Encryptor\encryptConfigs.bat*: Script for encrypting all sections of the configuration file.
- *\Misc\EA.Config.Encryptor\decryptConfigs.bat*: Script for decrypting all sections of the configuration file.

Install Core Server

System requirements

To install Core Server, run *Axidian.Server-<version number>.x64.en-us.msi*.

HTTPS binding in IIS Manager

Core Server is a web application that runs on IIS. During installation, the mandatory SSL requirement is enabled by default in the settings, which in turn requires an enabled HTTPS binding.

1. Start IIS Manager and expand the **Sites** item.
2. Select *Default Web Site* and in the **Actions** section, click **Bindings**.
3. Click **Add**:
   - *Type*: https.
   - *Port*: 443.
   - Select *SSL Certificate*.
4. Save the binding.

If you do not plan to use the HTTPS protocol, disable the SSL requirement in the IIS settings for Core and in *C:\inetpub\wwwroot\am\core\Web.config* change the value of the *requireHttps* parameter to *false*.

Example

```
<appSettings>
<add key="requireHttps" value="false" />
</appSettings>
```

Configuration wizard

The configuration wizard runs automatically by default after you install Core Server, unless you abort the wizard start.

To launch the wizard manually, execute *C:\Program Files\Axidian\Wizard\EA.Server.Wizard.exe*.

> ⓘ **INFO**
> The wizard includes automatic validation of entered data. If the data is correct, the field becomes green, and you can proceed to the next step. If the data is entered incorrectly, the fields are highlighted in red, and you cannot proceed to the next step until you specify correct data.

The following steps vary depending on the database used.

Microsoft SQL

Microsoft SQL database

1. On the **Before You Begin** step, click **Next**.
2. On the **Restore Settings** step, click **Next**.
3. On the **User Catalog** step, click **Add** and specify the following parameters:
    - *Domain name (FQDN)*: Fully qualified domain name of the Active Directory domain where the catalog is located, such as *domain.local.*
    - *Service account*: Service account that has access rights to the user catalog. Click **Edit** and specify the account credentials.
    - *LDAP path to catalog*: Path to catalog with users. Click **Select** and choose a container or the entire domain

    Click **Configure attribute mapping** to set the mapping between Active Directory and Axidian Access attributes.

Attribute list

- FirstName: User first name.
- MiddleName: User middle name.
- LastName: User last name.
- Email: User email address.
- Phone: User phone number.
- Name: User display name.
- CanonicalName: User canonical name, such as *domain.com/Users/testuser*.
- PrincipalName: User unique name in a specific format, such as *testuser@domain.com*.
- SamCompatibleName: User name compatible with the Security Account Manager (SAM) system. For example, *domain\testuser*.
- DistinguishedName: Unique name of an object in the Active Directory hierarchy consisting of the full path to the object. For example, *CN=testuser,OU=Users,DC=domain,DC=com*.
- Sid: Unique security identifier assigned to each object in Active Directory. Sid is used to manage access to resources and authentication of users and objects.

Click **Configure catalog object filtering** to set search criteria for records in user catalogs.

List of filters

- LDAP filter: Filter in the form of an LDAP query applied when accessing the user catalog. For more information, see LDAP filter when searching for users and groups in Active Directory.
- DN filter: The distinguishedName filter in the form of a regular expression used to filter results after executing an LDAP query.

4. On the **Data Storage** step, select the Microsoft SQL storage type.
5. In the **Database connection** string, click **Edit** to specify the data for connecting to the database that will be used as the Axidian Access data storage.
   - *Server name*: Database server name.

- Select the **SQL Server Authentication** authentication method and specify the service user credentials with full permissions for the database.
- In the **Select or enter a database name** section, select or specify the database that will be used as the Axidian Access storage.
- After completing the setting, click **Test Connection**. If the configuration is correct, a window with *Test connection succeeded* appears.
- Click **Ok** in all windows.

6. Step **Encryption Key**. Select the encryption algorithm, click **Generate** and then click **Next**.

> **TIP**
> We recommend you to create a backup of the encryption key and save it in a secure location.

7. On the **Log Server** step, specify the following data:

> **NOTE**
> To avoid errors when testing the connection, a fully configured Log Server with a configured database or installed EventLog component is required. If the log server is not ready for operation, you can skip this step.

- *Log Server address*: URL for connecting to the server in the format *http(s)://full_dns_server_name/ls/*, such as *http://logserver.demo.local/ls/*.
- *Certificate*: Certificate for configuring two-way TLS connection between Core Server and Log Server.

> **IMPORTANT**
> This setting is optional. Current version of Axidian Access does not support two-way TLS connection.

- *Logged catalog object field*: Specify the format in which the username will be logged.
- *Logged computer field*: Specify the format in which the *Computer* field will be logged.

> **NOTE**
> For logging in DNS format, additional configuration is required.

8. **Session secret**. To generate a key pair for signing the user token, click **Generate**.

The public key is saved in the *SessionPublicKey.pub* file. Add this key to all computers with Windows Logon or ESSO Agent installed. Use the *Session public key settings* policy located in *Axidian version number\Misc\ADMX Templates*.

9. **File encryption**. You can perform encryption of the configuration file settings.

> **TIP**
> For security purposes, we recommend you to encrypt the configuration file.

10. **Confirmation**. Verify the correctness of the specified data and click **Apply**.

> **TIP**
> We recommend you to make a backup of the configuration file, by default the option *Save backup of configuration parameters* is active.

11. **Results**. Settings verification and connection testing to Core Server is performed.

System administrator setting

> **NOTE**
> A user specified as a system administrator must be in the user catalog.

1. On the **System Administrator** step, specify the service account of the Axidian Access administrator. The specified account will be granted primary system administrator rights.

> **NOTE**
> For administrator setting, install Windows Password (*Axidian version number\Axidian Providers\Windows Password Provider\version number*), since when granting initial rights to the specified user, authentication is performed on Core Server.

2. Verify the correctness of the specified data and click **Apply**.

3. The **Results** step displays the status of the configuration file setting and the status of the system administrator setting.

PostgreSQL

PostgreSQL database

1. On the **Before You Begin** step, click **Next**.

2. On the **Restore Settings** step, click **Next**.

3. On the **User Catalog** step, click **Add** and specify the following parameters:

    - *Domain name (FQDN)*: Fully qualified domain name of the Active Directory domain where the catalog is located, such as *domain.local.*
    - *Service account*: Service account that has access rights to the user catalog. Click **Edit** and specify the account credentials.
    - *LDAP path to catalog*: Path to catalog with users. Click **Select** and choose a container or the entire domain.

    Click **Configure attribute mapping** to set the mapping between Active Directory and Axidian Access attributes.

Attribute list

- FirstName: User first name.
- MiddleName: User middle name.
- LastName: User last name.
- Email: User email address.
- Phone: User phone number.
- Name: User display name.
- CanonicalName: User canonical name, such as *domain.com/Users/testuser*.
- PrincipalName: User unique name in a specific format, such as *testuser@domain.com*.
- SamCompatibleName: User name compatible with the Security Account Manager (SAM) system. For example, *domain\testuser*.
- DistinguishedName: Unique name of an object in the Active Directory hierarchy consisting of the full path to the object. For example, *CN=testuser,OU=Users,DC=domain,DC=com*.
- Sid: Unique security identifier assigned to each object in Active Directory. Sid is used to manage access to resources and authentication of users and objects.

Click **Configure catalog object filtering** to set search criteria for records in user catalogs.

List of filters

- LDAP filter: Filter in the form of an LDAP query applied when accessing the user catalog. For more information, see LDAP filter when searching for users and groups in Active Directory.
- DN filter: The distinguishedName filter in the form of a regular expression used to filter results after executing an LDAP query.

4. On the **Data Storage** step, select the PostgreSQL storage type.

5. In the **Database connection string** field, specify the following data:

   - *server*: IP address of the database server.
   - *port*: The port used. The default value is 5432.
   - *user id*: The name of the user who has all rights to the database.

- *password*: User password.
- *database*: The database containing data about Core Server.

Click **Test** to check the database connection.

6. Step **Encryption Key**. Select the encryption algorithm, click **Generate** and then click **Next**.

> **TIP**
> We recommend you to create a backup of the encryption key and save it in a secure location.

7. On the **Log Server** step, specify the following data:

> **NOTE**
> To avoid errors when testing the connection, a fully configured Log Server with a configured database or an installed EventLog component is required. If the log server is not ready for operation, you can skip this step.

- Log server address: URL for connecting to the server in the format *http(s)://full_dns_server_name/ls/*, such as *http://logserver.demo.local/ls/*.
- *Certificate*: Certificate for configuring two-way TLS connection between Core Server and Log Server.

> **IMPORTANT**
> This setting is optional. The current version of the Axidian Access system does not support two-way TLS connection operation.

- *Logged catalog object field*: Specify the format in which the username will be logged.
- *Logged computer field*: Specify the format in which the *Computer* field will be logged.

> **NOTE**
> For logging in DNS format, additional setting is required.

8. **Session secret**. To generate a public key for token signing, click **Generate**.

   The public key is saved in the *SessionPublicKey.pub* file. Add this key to all computers with Windows Logon or ESSO Agent installed. Use the *Session public key settings* policy located in *Axidian version number\Misc\ADMX Templates*.

9. **File encryption**. You can perform encryption of the configuration file settings .

> **TIP**
> For security purposes, we recommend you to encrypt the configuration file.

10. **Confirmation**. Verify the correctness of the specified data and click **Apply**.

> **TIP**
> We recommend you to make a backup of the configuration file, by default the option *Save backup of configuration parameters* is active.

11. **Results**. Settings verification and connection testing to Core Server is performed.

System administrator setting

> **INFO**
> A user specified as a system administrator must be in the user catalog.

1. On the **System Administrator** step, specify the service account of the Axidian Access administrator. The specified account will be granted primary system administrator rights.

> **INFO**
> To configure the administrator, install Windows Password must be installed, as authentication on the Axidian Access server is performed when granting primary rights to the s pecified user

2. **Confirmation**. Verify the correctness of the specified data and click **Apply**.
3. The **Results** step displays the status of the configuration file setting and the status of the system administrator setting.

Manual configuration

> **INFO**
> We recommend you to edit the configuration file with regard to the peak server load.

To save changes to the configuration file, run the editor as administrator.

Errors that occur during Core Server installation (for example, an error in the configuration file) are logged based on the Log Server settings.

Requirements for characters in the password for service users

If the following characters are used in the password for service users: **&**, **"**, **;**, , **space**, then the password should be specified as follows:

- Ampersand character (&): When specifying the password for a user from AD or SQL, the character must be replaced with the **&** combination Example (AD\SQL): **password="Q1q2E3e4&"\Password='Q1q2E3e4&';**
- The double quote symbol (''): When specifying a password for a user from AD or SQL, the symbol must be replaced with the **"** combination. For a user from SQL, it is additionally required to specify the password in single quotes. Example(AD\SQL): **password="Q1q2E3e4""\Password='Q1q2E3e4"';**
- The semicolon symbol (;): When specifying a password for a user from AD, no additional actions are required. For a user from SQL, the password must be specified in single quotes. Example(AD\SQL): **password="Q1q2E3e4;"\Password='Q1q2E3e4;';**
- The less than symbol (<): When specifying a password for a user from AD or SQL, the symbol must be replaced with the **<** combination. Example(AD\SQL): **password="Q1q2E3e4<"\Password='Q1q2E3e4<';**
- The space symbol: When specifying a password for a user from AD, no additional actions are required; for a user from SQL, the password must be specified in single quotes. Example(AD\SQL): **password="Q1q2E3e4 "\Password='Q1q2E3e4';**
- The single quote symbol ('): When specifying a password for an AD user, no additional actions are required; for an SQL user, the password must be specified as the **"** combination, the quote symbol should be specified in its regular form. Example(AD\SQL): **password="Q1q2E3e4""\Password="Q1q2E3e4'"**

Microsoft SQL

1. Open *C:\inetpub\wwwroot\am\core\Web.config* for editing.
2. In the `sessionEncryptionSettings` parameter, specify the public and secret keys for signing the user token. To generate keys, run the *AccessManager.KeyGen.Console.exe* tool located in the *Axidian Access Manager Server/version number/Misc* catalog.

In the `--name` parameter, specify the file names. This is a required parameter. Files are generated with the same name. However, the public key file has the PUB extension.

In the `--output-directory` parameter, specify the path to the catalog where the key pair files are saved. You can specify a relative or absolute path. This is an optional parameter. If you do not specify a path, the key pair files are saved in the *Misc* catalog.

3. To specify the system user catalog, edit the following parameters in the `adUserCatalogProvider` tag:

   - `id`: Arbitrary unique catalog identifier.
   - `serverName`: Active Directory domain name in which the catalog is located.
   - `containerPath`: Path to the container in the Distinguished Name format or the entire domain if the entire domain is used for storing users.
   - `userName`: Name of the service account for connecting to the user catalog.
   - `password`: Password of the service account in the Active Directory user catalog.

   Example

   ```
   <adUserCatalogProviders>
       <adUserCatalogProvider id="UserId" serverName="axidian.local"
   containerPath="DC=demo,DC=local" userName="AxidianCatalogUser"    password="Q1q2E3e4"/>
   </adUserCatalogProviders>
   ```

4. To specify the root user catalog provider identifier, edit the `rootUserCatalogProviderId` attribute in the `userCatalogProviderSettings` tag.

   - `rootUserCatalogProviderId` : Set the value that has already been set in the tag `adUserCatalogProvider` in the attribute `id`.

   Example

   ```
   <userCatalogProviderSettings rootUserCatalogProviderId="UserId">
   ```

5. Set the system data storage. For data storage in SQL Server, edit the tag `dbContextSettings` and create the tag `mssqlDbContext` with parameters `id` and `connectionString`.

   - `rootDbContextId` : Set an arbitrary unique value for the data storage identifier.
   - `id`: Set the value that is set in the tag `rootDbContextId`.

6. Add the `connectionString` parameter with built-in parameters:

- `Data Source`: Specifies the server instance. This property is required for all connections. Valid values include the network name or IP address of the server, local or localhost for local connections.
- `Initial Catalog`: Specifies the database name.
- `User Id`: User name for connecting to the database.
- `Password`: User password for connecting to the database.

Example

```
<dbContextSettings rootDbContextId="mssql"> <mssqlDbContexts> <mssqlDbContext id="mssql" connectionString="Data Source=EASERVER\EASERVER;Initial Catalog=AM_Server_7;User Id=Admin-DB;Password=Q1q2E3e4;"/> </mssqlDbContexts> </dbContextSettings>
```

7. Set the system data encryption key. Edit the parameters in the tag `encryptionSettings`.

   - `cryptoAlgName`: Specify the encryption algorithm used. Possible values:
     - DES
     - TripleDES
     - RC2
     - AES
     - Rijndael
   - `cryptoKey`: Key value generated by the utility.
   - `certificateThumbprint`: Certificate thumbprint used to encrypt the key (to ignore, remove the attribute).

   Example

   ```
   <encryptionSettings cryptoAlgName="Aes" cryptoKey="90ce7dbc3ff94a7867abc6672c23cce2c3717d38af42f04293130cb68a34ecc2"/>
   ```

8. Set the system administrator. Edit the `userId` parameter of the `accessControlAdminSettings` tag.

   - `userId`: User identifier in the format *<Catalog identifier><underscore><System Administrator GUID>*.

   > **INFO**
   >
   > The user must be located in the user catalog.
   >
   > When using multiple user containers, the *Catalog identifier* specifies the container id where the system administrator is located.

Example

```
<accessControlAdminSettings userId="UserId_84e9ccd9-73a2-43c7-abc6-604a16902037"/>
```

> **INFO**
> To obtain the GUID, use a PowerShell command. You must first install the Remote Server Administration Tools component:
>
> **Example**
>
> ```
> Get-ADUser YouUserName -Properties * | Select ObjectGUID
> ```

9. Set the url to connect to Log Server. Edit the `logServer` tag.
   - `URL`: The url address to connect to Log Server in the format http(*s*)*://server name/ls/api.*

> **NOTE**
> If you use multiple servers, specify the load balancer address.

   - `CertificateThumbprint`: If the private key is in the registry and the certificate is in the computer certificate store.
   - `CertificateFilePath`: If the key pair is in pfx.
   - `CertificateFilePassword`: Password for the pfx.

Example

```
<logServer Url="http://log.axidian.local/ls/api/" CertificateThumbprint=""
CertificateFilePath="" CertificateFilePassword=""/>
```

Encrypt/Decrypt the configuration file

1. Run the command prompt as administrator.
2. In the command prompt, navigate to the folder with the encryption utility.

> **TIP**
> The utility encrypts the `logServer`, `logonSettings`, `userCatalogProviderSettings`, `encryptionSettings`, `dbContextSettings` sections. We recommend you to encrypt all sections.

- Encrypt/Decrypt individual sections. To encrypt an individual section, execute a command, such as:

```
EA.Config.Encryptor /encrypt "Path to server configuration file Section name"
```

Command example

```
EA.Config.Encryptor /encrypt "C:\inetpub\wwwroot\am\core\Web.config" "logServer"
```

- To decrypt an individual section, execute the following command:

```
EA.Config.Encryptor /decrypt "Path to server configuration file" "Section name"
```

Command example

```
EA.Config.Encryptor /decrypt "C:\inetpub\wwwroot\am\core\Web.config" "logServer"
```

- Encrypt/Decrypt all sections. To encrypt all sections, run the script `encryptConfigs.bat`.
- To decrypt all sections, execute the script `decryptConfigs.bat`.

PostgreSQL

1. Open *C:\inetpub\wwwroot\am\core\Web.config* for editing.
2. In the `sessionEncryptionSettings` parameter, specify the public and secret keys for signing the user token. To generate keys, run the *AccessManager.KeyGen.Console.exe* tool located in the *Axidian Access Manager Server/version number/Misc* catalog.

   In the `--name` parameter, specify the file names. This is a required parameter. Files are generated with the same name. However, the public key file has the PUB extension.

   In the `--output-directory` parameter, specify the path to the catalog where the key pair files are saved. You can specify a relative or absolute path. This is an optional parameter. If you do not specify a path, the key pair files are saved in the *Misc* catalog.
3. To specify the system user catalog, edit the following parameters in the `adUserCatalogProvider` tag:

- `id`: Arbitrary unique catalog identifier.
- `serverName`: Active Directory domain name in which the catalog is located.
- `containerPath`: Path to the container in the Distinguished Name format or the entire domain if the entire domain is used for storing users.
- `userName`: Name of the service account for connecting to the user catalog.
- `password`: Password of the service account in the Active Directory user catalog.

Example

```
<adUserCatalogProviders>
    <adUserCatalogProvider id="UserId" serverName="axidian.local"
containerPath="DC=demo,DC=local" userName="AxidianCatalogUser"    password="Q1q2E3e4"/>
</adUserCatalogProviders>
```

4. To specify the root user catalog provider identifier, edit the `rootUserCatalogProviderId` attribute in the `userCatalogProviderSettings` tag.

   - `rootUserCatalogProviderId`: Set the value that is set in the `adUserCatalogProvider` tag in the `id` attribute.

   Example

   ```
   <userCatalogProviderSettings rootUserCatalogProviderId="UserId">
   ```

5. Set the system data storage. For data storage in PostgreSQL Server, edit the `dbContextSettings` tag and create a `postgresqlDbContexts` tag with `id` and `connectionString` parameters.

   - `rootDbContextId`: Set an arbitrary unique value for the data storage identifier.
   - `id`: Set the value that is set in the `rootDbContextId` tag.

6. Add the `connectionString` parameter with built-in parameters:

   - `Server`: DNS\IP address of the server with the PostgreSQL database.
   - `Database`: Defines the database name.
   - `Integrated Security`: Enables built-in authentication.

   > **INFO**
   > Built-in authentication is not supported.

   - `User ID`: Username for connecting to the database.
   - `Password`: User password for connecting to the database.

Example

```
<dbContextSettings rootDbContextId="postgreSql">
    <postgresqlDbContexts>
      <postgresqlDbContext connectionString="server=192.168.1.2;port=5432;user
id=postgres;password=!QAZ2wsx;database=axidiancore"
        id="postgreSql" />
    </postgresqlDbContexts>
</dbContextSettings>
```

7. Set the system data encryption key. Edit the following parameters in the `encryptionSettings` tag.

   - `cryptoAlgName`: Specify the encryption algorithm used. Possible values:
     - DES
     - TripleDES
     - RC2
     - AES
     - Rijndael
   - `cryptoKey`: Key value generated by the utility.
   - `certificateThumbprint`: Certificate thumbprint used to encrypt the key (to ignore, remove the attribute).

Example

```
<encryptionSettings cryptoAlgName="Aes"
cryptoKey="90ce7dbc3ff94a7867abc6672c23cce2c3717d38af42f04293130cb68a34ecc2"/>
```

8. Set the system administrator. Edit the `userId` parameter of the `accessControlAdminSettings` tag.

   - `userId`: User identifier in the format *Catalog identifier_underscore/_System Administrator guid*.

> **NOTE**
> The user must be located inside the user catalog.
>
> When using multiple user containers, specify the container id where the system administrator is located for *Catalog identifier*.

Example

```
<accessControlAdminSettings userId="UserId_84e9ccd9-73a2-43c7-abc6-604a16902037"/>
```

> **INFO**
>
> To obtain the GUID, use a PowerShell command. You must first install the Remote Server Administration Tools component:
>
> **Example**
>
> ```
> Get-ADUser YouUserName -Properties * | Select ObjectGUID
> ```

9. Set the url to connect to Log Server. Edit the `logServer` tag.
    - `URL`: The url address to connect to Log Server in the format *http(s)://server name/ls/api*.

> **NOTE**
>
> If you use multiple servers, specify the load balancer address.

    - `CertificateThumbprint`: If the private key is in the registry and thecertificate is in the computer certificate store.
    - `CertificateFilePath`: If the key pair is in pfx.
    - `CertificateFilePassword`: Password for the pfx.

Example

```
<logServer Url="http://log.axidian.local/ls/api/" CertificateThumbprint=""
CertificateFilePath="" CertificateFilePassword=""/>
```

Encrypt/Decrypt the configuration file

1. Run the command prompt as administrator.
2. In the command prompt, navigate to the folder with the encryption utility.

> **TIP**
>
> The utility encrypts the `logServer`, `logonSettings`, `userCatalogProviderSettings`, `encryptionSettings`, `dbContextSettings` sections. We recommend you to encrypt all sections.

- Encrypt/Decrypt individual sections. To encrypt an individual section, execute a command of the form:

```
EA.Config.Encryptor /encrypt "Path to server configuration file" "Section name"
```

Example

```
EA.Config.Encryptor /encrypt "C:\inetpub\wwwroot\am\core\Web.config" "logServer"
```

- To decrypt an individual section, execute a command of the form:

```
EA.Config.Encryptor /decrypt "Path to server configuration file" "Section name"
```

Example

```
EA.Config.Encryptor /decrypt "C:\inetpub\wwwroot\am\core\Web.config" "logServer"
```

- Encrypt/Decrypt all sections. To encrypt all sections, run the `encryptConfigs.bat` script.
- To decrypt all sections, execute the `decryptConfigs.bat` script.

Check the server status

To check the server status, enter the Core Server address in your browser in the format *https://full_dns_server_name/am/core/*.

You are automatically redirected to *https://full_dns_server_name/am/core/api/v6/healthCheck/index*.

The page refreshes automatically every 60 seconds.

The page provides the following information about the server status:

- The availability status of the user catalog and the processing time of requests to it.
- The storage status and the processing time of requests to it.
- The list of installed authentication providers, their status, and load time.
- The status of the Log Server component.
- The list of errors, if any.

> **NOTE**
> For security reasons, the list of installed authentication providers and the error information are available only if you have opened the page from a local address.

Enable brute-force protection

In Axidian Access, you can configure protection against account brute-forcing for the Identity Provider component.

If this setting is enabled, when a non-existent username is entered, Axidian Access simulates the login of an existing user through Identity Provider: it displays the authentication methods, requests a password, and then displays the error *Invalid username or authenticator, or the device is blocked*.

If this setting is disabled, when a non-existent username is entered, Axidian Access displays the error *Internal server error: User not found*.

By default, this setting is disabled.

To configure brute-force protection, perform the following actions:

1. Open the Core Server configuration file *web.config* located in *C:\inetpub\wwwroot\am\core*.
2. Modify the *appSettings* parameter. In the *bruteForceProtectionApps* line, specify the modules for which you want to enable brute-force protection, separated by commas.

Example

```
<appSettings>
    <add key="bruteForceProtectionApps" value="Identity Provider, Windows Logon, IIS Extension"
/>
    ...
</appSettings>
```

3. Save the changes and restart IIS.

Configure multiple custom directories

If several directories are configured in the Axidian Access system that is already in use (after granting primary rights to the system administrator), and the location of the system administrator or the prefix specified in the `accessControlAdminSettings` parameter changes, you must delete the previously granted rights and restart the primary configuration utility.

To delete the rights, you must delete all data from the *DbAccessGroupMembers* table located in the database of the system.

> **TIP**
> If the containers are located in different domains/forests, you must create a user to read data from the container in your domain/forest.

1. Add lines to the `adUserCatalogProviders` tag to connect to containers.

   Example

```
<userCatalogProviderSettings rootUserCatalogProviderId="user">
 <userCatalogProviders>
   <sqlUserCatalogProviders></sqlUserCatalogProviders>
   <adUserCatalogProviders>
     <adUserCatalogProvider id="Ad1" serverName="demo.local"
containerPath="OU=AxidianAccess_Users,DC=demo,DC=local" userName="demo\ind-user"
password="Q1q2E3e4" />
     <adUserCatalogProvider id="Ad2" serverName="inforest.demo.local"
containerPath="OU=UsersInForest,DC=inforest,DC=demo,DC=local"
userName="inforest\cataloguser1" password="Q1q2E3e4" />
     <adUserCatalogProvider id="Ad3" serverName="newforest.local"
containerPath="OU=Usersoutforest,DC=newforest,DC=local" userName="newforest\cataloguser2"
password="Q1q2E3e4" />
   </adUserCatalogProviders>
 </userCatalogProviders>
```

2. Add the `orUserCatalogProvider` tag with the `id` parameter inside the `orUserCatalogProviders` tag.

> **INFORMATION**
> The value of the `id` parameter must match the value specified in the `rootUserCatalogProviderId` parameter.

3. Add the `userCatalogProviders` tag inside the `orUserCatalogProvider` tag. Inside the `userCatalogProviders` tag, add the `userCatalogProvider` tags with the `id` parameter, which

specifies the ID of the user container, and `ignoreExceptions` with the value `true`. This parameter ignores the error connecting to the directory if this directory is unavailable.

> **INFORMATION**
>
> These tags may not be present in the configuration file if the configuration file has previously been encrypted with unspecified parameters. If there are no tags, then add them manually. The full file structure is represented below.

Example

```
<orUserCatalogProviders>
    <orUserCatalogProvider id="user">
       <userCatalogProviders>
          <userCatalogProvider id="Ad1" ignoreExceptions="true" />
          <userCatalogProvider id="Ad2" ignoreExceptions="true" />
          <userCatalogProvider id="Ad3" ignoreExceptions="true" />
       </userCatalogProviders>
    </orUserCatalogProvider>
</orUserCatalogProviders>
```

Example of the file structure

```
<accessControlAdminSettings userId="UserId_891f2b6c-9a55-4e1a-b69b-b4d6418f4c4c"/>
  <logonSettings secretKey="*****"/>
  <userCatalogProviderSettings rootUserCatalogProviderId="user">
    <userCatalogProviders>
        <sqlUserCatalogProviders>
        </sqlUserCatalogProviders>
        <adUserCatalogProviders>
            <adUserCatalogProvider id="UserId" serverName="new.loc"
containerPath="DC=new,DC=loc"
                userName="axidian-users" password="Q1q2E3e4" />
            <adUserCatalogProvider id="UserId1" serverName="test.new.loc"
containerPath="DC=test,DC=loc"
                userName="axidian-users" password="Q1q2E3e4" />
        </adUserCatalogProviders>
    </userCatalogProviders>
    <combineRules>
        <orUserCatalogProviders>
            <orUserCatalogProvider id="user">
               <userCatalogProviders>
                  <userCatalogProvider id="UserId" ignoreExceptions="true" />
                  <userCatalogProvider id="UserId1" ignoreExceptions="true" />
               </userCatalogProviders>
            </orUserCatalogProvider>
        </orUserCatalogProviders>
        <andUserCatalogProviders>
        </andUserCatalogProviders>
    </combineRules>
  </userCatalogProviderSettings>
```

Configure LDAP filters in Active Directory search

If you need to exclude duplicate Active Directory (AD) accounts when searching for users and groups, or if you need to access only part of the directory of users, and not the entire directory, you can configure the LDAP filter as described.

1. On Core Server, open the server configuration file *C:\inetpub\wwwroot\am\core\Web.config*.
2. Find the `adUserCatalogProvider` section and set the `catalogFilter` and `dnFilter` attributes in it for filtering user catalog items:
    1. `catalogFilter` is an LDAP query string for filtering items at the stage of executing the request to AD. It does not support filtering based on the partial match of the `distinguishedName` and `canonicalName` attributes (`dnFilter` is for this purpose).
    2. `dnFilter` (`distinguishedName` filter) is a regular expression for filtering results after executing an LDAP request to AD.

Examples of using filters

catalogFilter

- exclusion of certain users `&(!(name=AmUser1))(!(name=AmUser2)(!(name=AmUser3)))`
- exclusion of all child elements of the first level of the specified OU `!(msDS-parentdistname=OU=AmOrgUnit,OU=AmUserCatalog,DC=dom,DC=local)`
- exclusion of all members of the group `!(memberOf=CN=AmGroup,OU=AmUserCatalog,DC=dom,DC=local)`

dnFilter

- exclusion of OU and all child elements `(?!^.*OU=MyOULevel1,OU=AmUserCatalog,DC=dom,DC=local$)(^.*$)`
- using only the specified OU `(^.*OU=MyOULevel1,OU=AmUserCatalog,DC=dom,DC=local$)`
- combined filter — using only the specified OU without defined child elements `(?!.*OU=MyOULevel2-1,OU=MyOULevel1,OU=AmUserCatalog,DC=dom,DC=local$)(^.*OU=MyOULevel1,OU=AmUserCatalog,DC=dom,DC=local$)`

- multi-OU filtering design via the | (or) operator — any number of conditions can be met using the full `distinguishedName` of the specified OU `((^.*OU=Axidian3,OU=AxidianUnit,DC=axidian,DC=local$)|(^.*OU=Axidian4,OU=AxidianUnit,DC=axidian,DC=local$))`

Configure license recall mechanism

Axidian Access uses an automatic license release mechanism. This mechanism is enabled by default and checks once a day (every 86,400 seconds) whether the conditions for license revocation are met.

A license is revoked from a user in the following cases:

- The user is removed from a policy.
- The user is disabled or moved out of the policy scope.
- The user is blocked or deleted from the user catalog.

If the license revocation mechanism is disabled, licenses are not released even when these conditions are met.

To change the time after which the license is revoked:

1. Open the configuration file *C:\inetpub\wwwroot\am\core\Web.config*.
2. Make the following changes in the `cleanLicense` tag:
    - For the `intervalSec` parameter, specify the required value in seconds. It is not recommended to set the value to less than 600 seconds.
    - For the `enabled` parameter, set the value to `true`.
3. Save the changes.
4. Restart IIS. The changes made are applied after the restart.

> **NOTE**
> We recommend you to edit the configuration file and restart IIS with regard to the peak load on the server.

If the licenses are not revoked automatically, restart the license revocation mechanism as follows:

1. Open the configuration file *C:\inetpub\wwwroot\am\core\Web.config*.
2. In the `cleanLicense` tag, set the `enabled` parameter to `false`.
3. Save the changes.
4. Restart IIS.

5. In the *Web.config* file, in the `cleanLicense` tag, in the `enabled` parameter, set the value to `true`.

6. Save the changes and restart IIS.

> **NOTE**
>
> Based on the results of license revocation, an entry with the code 1094 is created in the event log. The event record contains the total number of revoked licenses.

Management Console

Axidian Management Console (Management Console) is a web application that runs on the basis of IIS. It allows you to administer the system and user settings.

> ⓘ **HOW TO OPEN MANAGEMENT CONSOLE**
>
> - In the browser: *http(s)://server dns/am/mc*
> - In IIS: <machine *name*>*sites**Default Web Site**am**mc*

Set up Management Console:

1. Install the module.
2. Create an HTTPS binding in the IIS settings.
3. Set up authentication.
4. If required, configure optional settings.

Install Management Console

System requirements

1. To install Management Console, run *Axidian.ManagementConsole-version number.x64.en-us.msi* located at *Axidian version number**Axidian Management Console version number*.

2. When the installation is complete, you will be prompted to generate a new IDP certificate. This is an optional setting.

   > ⓘ **INFO**
   >
   > The certificate is required for authentication in Management Console using Identity Provider. This certificate is not intended for establishing the SSL connection.
   >
   > If this certificate was generated earlier, disable the option.
   >
   > If this option is enabled, a new self-signed certificate is generated and installed in the *Local Machine* → *Personal* certificate store. The certificate is used to encrypt data transmitted between the authentication server and the client application.

Create HTTPS binding

If during the installation the requirement for SSL certificate is enabled in the IIS settings, you must create an HTTPS binding as follows:

1. In IIS Services Manager, in the left menu, navigate to **Axidian Access** → **sites** → **Default Web Site**.
2. In the **Actions** section of the right menu, click **Bindings**.
3. Click **Add** and set the following parameters in the window that appears:
   - In the **Type** field, select *https*.
   - In the **Port** field, specify *443*.
   - In the **SSL Certificate** field, select the Core Server certificate.
4. Click **OK** to save the link.

If you do not plan to use the HTTPS protocol, disable the SSL certificate requirement in the IIS settings for Management Console.

Also, in *C:\inetpub\wwwroot\am\mc\Web.config*, change the value of `the requireSSL` parameter to `false` as follows:

```
<httpCookies httpOnlyCookies="true" requireSSL="false" />
```

Configure authentication

You can set different authentication methods to log in to Management Console.

Windows Authentication

Authentication using Windows Authentication

To set up authentication using Windows Password, enable Windows Authentication and edit the console configuration file.

Enable Windows authentication

1. Open IIS Services Manager and in the left menu, expand the **Axidian Access** → **sites** → **Default Web Site** → **am** → **mc** node.
2. Click **Authentication**.

3. Turn on the **Impersonation ASP.NET and Windows** authentication. Disable all other authentication methods.

Modify the configuration file

1. Open *C:\inetpub\wwwroot\am\mc\Web.config* for editing.
2. In the `amAuthServer` section, in the `Url` parameter, specify the address for connecting to Core Server, such as *http(s)://full_dns_name of the server/am/core/*.

```
<amAuthServer Url="https://amcore.test.local/am/core/">
```

3. In the `LogServer` section, in the `Url` parameter, specify the address for connecting to Log Server, such as *http(s)://full_dns_name of the server/ls/api*.

```
<logServer Url="http://amcore.test.local/ls/api/">
```

If multiple servers are used, specify the address of the load balancer.

4. Save the changes and restart the IIS server.

Management Console will be available at: *http(s)://full_dns_name of the server/am/mc/*.

> ⓘ **NOTE**
> To ignore server certificate errors when logging in to the console, in *C:\inetpub\wwwroot\am\mc\Config\applicationSettings.config* change the value of the `isIgnoreCertErrors` parameter to `true`.

Identity Provider

Authentication using Identity Provider

To set up a secure login to Management Console using Identity Provider:

1. Install and configure the Identity Provider module.
2. Install and configure an authentication provider.
3. Set up authentication in IIS Service Manager.
4. Edit the Management Console and Identity Provider configuration files.

You can also configure logging out of Management Console using Identity Provider.

Set up authentication

1. In IIS Services Manager, in the left menu, expand the **Axidian Access** → **sites** → **Default Web Site** → **am** → **mc** node.
2. Click **Authentication**.
3. Enable options **Anonymous authentication** and **Authentication using forms**. Disable all other authentication methods.

Change the Management Console configuration file

1. Open *C:\inetpub\wwwroot\am\mc\Web.config* for editing.
2. In the `amAuthServer` section, in the `Url` parameter, specify the address for connecting to Core Server, such as *http(s)://full_dns_name of the server/am/core/*.

```
<amAuthServer Url="https://amcore.test.local/am/core/">
```

3. In the `LogServer` section, in the `Url` parameter, specify the address for connecting to Log Server, such as *http(s)://full_dns_name of the server/ls/api*.

```
<logServer Url="http://amcore.test.local/ls/api/">
```

If multiple servers are used, specify the address of the load balancer.

4. Make the following changes in the `amAuthentication` section:
    - In the `mode` parameter, specify the `Saml` value.
    - In the `loginUrl` parameter, specify the Identity Provider address, such as *http(s)://full_dns_name of the server/am/idp/*.

      ```
      <amAuthentication mode="Saml" loginUrl="https://amcore.test.local/am/idp/>
      ```

    - In the `identityProviderCertificateThumbprint` parameter, specify the fingerprint of the Identity Provider certificate.

      ```
      Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=idp"}
      ```

> **TIP**
> If Identity Provider and Management Console are installed on different servers, export the certificate without the private key from the server with the Identity Provider module to the server with Management Console. Add the certificate to the certificate store in the *Local Machine* → *Personal* section.

- In the `serviceProviderCertificateThumbprint` parameter, specify the fingerprint of the Management Console certificate that was generated during the installation. The certificate is installed in the *Local Machine* → *Personal* certificate store with the common name `mcsp`.

```
Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=mcsp"}
```

5. Save the changes and restart IIS.

Management Console will be available at: *http(s)://full_dns_name of the server/am/mc/*.

> **NOTE**
> To ignore server certificate errors when logging in to the console, change the value of `the isIgnoreCertErrors` parameter to `true` in *the C file.:\inetpub\wwwroot\am\mc\Config\applicationSettings.config*.

Change the Identity Provider configuration file

1. Open *C:\inetpub\wwwroot\am\idp\app-settings.json* for editing.

2. In the `PartnerServiceProviderConfigurations` section, make the following changes:

   - In the `SingleLogoutServiceUrl` parameter, specify the address of the server with the Management Console component installed, such as *http(s):/full_dns_name of the server/am/mc/Account/SLOService*.

   - In the `Thumbprint` parameter, specify the fingerprint of the Management Console certificate that was generated during the installation of the package.

> **TIP**
>
> If Identity Provider and Management Console were installed on different servers, export the certificate without the private key from the server with Management Console to the server with the Identity Provider module. Add the certificate to the certificate store in the *Local Machine* → *Personal* section.

```
Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=mcsp"}
```

Example

```
   "PartnerServiceProviderConfigurations":
               "Name": "urn:axidianid:emc",
               //highlight-grey-next-line
               "SingleLogoutServiceUrl":
"https://server.axidian.local/am/mc/Account/SLOService",
               "PartnerCertificates":
                   //highlight-grey-next-line
                   "Thumbprint": "C77EDF29EA05B468BDAF553DE3D804DA4B139C1E"
```

3. Save the changes and restart IIS.

Optional settings

Set the session lifetime

To change the session lifetime in Management Console, perform the following steps:

1. Open *C:\inetpub\wwwroot\am\mc\Config\applicationSettings.config*.
2. Set the required value for the `sessionExpirationTimeInMinutes` parameter. The default value is 30 (minutes).

```
<amApplicationSettings
  findUsersMaxResultCount="200"
  isIgnoreCertErrors="false"
  sessionExpirationTimeInMinutes="60"
  allowOverrideRandomPasswordGeneration="false"
/>
```

3. Save the changes.

Configure user search

You can configure user search in Management Console by the following parameters:

- By first name
- By last name
- By login
- By the user name in the Active Directory catalog
- By first name and last name

To configure the search:

1. Open *C:\inetpub\wwwroot\am\mc\Config\userSearchSettings.config*.
2. In the `searchTemplate` parameter, specify the search template. Examples of valid templates: **{0}, {0}*, *{0}**.
3. Enable or disable the search by the following parameters by setting the value to `true` or `false`:
   - `searchByGivenName`: Search by first name. The default value is `true`.
   - `searchBySn`: Search by last name. The default value is `true`.
   - `searchByUpn`: Search by login. The default value is `true`.
   - `searchByGivenNameAndSn`: Search by first name and last name. The default value is `true`.
   - `searchByName`: Search by the name displayed in the list of users in Active Directory. The default value is `true`.

For convenience, you can use templates:

Request example	Comment
john*	Depending on your settings, searches for a user whose first name, last name, login, first name and last name, or user name in AD starts with "john".
oh	Depending on your settings, searches for a user whose first name, last name, login, first name and last name, or user name in AD contains "oh" at the beginning, in the middle, or at the end.
*ohn	Depending on your settings, searches for a user whose first name, last name, login, first name and last name, or user name in AD ends with "ohn".

Request example	Comment
john	A request without * is converted to the template specified in the `searchTemplate` parameter. If `searchTemplate` is not specified, * is added to the request on both sides.
john ** john	Examples of invalid requests. They return no search results.

Display/hide version number in Management Console

To hide the information about the platform and the version the application was developed on from the API response, perform the following actions:

1. Open the Management Console configuration file *C:\inetpub\wwwroot\am\mc\Web.config*.
2. Depending on the version of IIS:
   - For IIS 10: In the `security` section, move the comment `<!--<requestFiltering removeServerHeader="true" >-->` from the second line to the third one.

     ```
     <security>
     <requestFiltering>
     <!--<requestFiltering removeServerHeader="true" >-->
         <requestLimits maxAllowedContentLength="104857600" />
     </requestFiltering>
     </security>
     ```

   - For IIS above version 10, perform the following actions:
     - Install the URL Rewrite module.
     - Add the following lines to the `system.webServer` section of *C:\inetpub\wwwroot\am\mc\Web.config*:

       ```
       <rewrite>
         <outboundRules>
           <rule name="replace server header" patternSyntax="Wildcard" lockItem="false">
             <match serverVariable="RESPONSE_SERVER" pattern="*" />
             <action type="Rewrite" value="MyServer" />
           </rule>
         </outboundRules>
       </rewrite>
       ```

3. Save the changes and restart IIS.

Configure logout using Identity Provider

To configure logout from Management Console using Identity Provider:

1. Open the console configuration file *C:\inetpub\wwwroot\am\mc\Web.config*.
2. In the `amAuthentication` section, add the `enableLogout` parameter with the `true` value.

```
<amAuthentication mode="Saml" loginUrl="https://amcore.test.local/am/idp/"
enableLogout="true"/>
```

3. Open the Identity Provider configuration file *app-settings.json* located in *C:\inetpub\wwwroot\am\idp*.
4. In the `PartnerServiceProviderConfigurations` section, in the `SingleLogoutServiceUrl` parameter, specify the address of the server with the Management Console component in the format *http(s)://full_dns_server_name/am/mc/Account/SLOService*.

```
"PartnerServiceProviderConfigurations": [
          {
            ...
            "SingleLogoutServiceUrl": "https://amcore.test.local/am/mc/Account/SLOService",
            ...
            ]
          }
```

5. Save the changes and restart IIS.

User Console

Axidian User Console (User Console) is a web application that runs on IIS. In User Console, the user can manage their authenticators.

> ⓘ **HOW TO OPEN USER CONSOLE**
>
> - In the browser: *http(s)://server dns/am/uc*
> - In IIS: <machine *name*>*sites**Default Web Site**am**uc*

Install User Console

System requirements

1. To install User Console, start *Axidian.UserConsole-version number.x64.en-us.msi* located at *Axidian version number**Axidian User Console**version number*.

2. When the installation is complete, you will be prompted to generate a new IDP certificate. This is an optional setting.

   > ⓘ **INFO**
   >
   > The certificate is required for authentication in User Console using Identity Provider. This certificate is not intended for establishing the SSL connection.
   >
   > If this certificate was generated earlier, disable the option.
   >
   > If this option is enabled, a new self-signed certificate is generated and installed in the *Local Machine* → *Personal* certificate store. The certificate is used to encrypt data transmitted between the authentication server and the client application.

HTTPS binding in IIS Manager

User Console is a web application that runs on IIS. During the installation process, SSL is required for User Console in the settings, which requires to enable HTTPS connection.

1. In IIS Services Manager, in the left menu, navigate to **Axidian Access** → **sites** → **Default Web Site**.

2. In the **Actions** section of the right menu, click **Bindings**.
3. Click **Add** and set the following parameters in the window that appears:
   - In the **Type** field, select *https*.
   - In the **Port** field, specify *443*.
   - In the **SSL Certificate** field, select the Core Server certificate.
4. Click **OK** to save the link.

If you do not plan to use the HTTPS protocol, disable the SSL requirement in the IIS settings for User Console.

Also, in *C:\inetpub\wwwroot\am\uc\Web.config*, change the value of the `requireSSL` parameter to `false` as follows:

```
<httpCookies httpOnlyCookies="true" requireSSL="false" />
```

Confirgure authentication

You can set different authentication methods to log in to User Console.

Windows Authentication

Authentication using Windows Authentication

By default, User Console is configured to use SAML authentication. If necessary, you can configure transparent Windows authentication as follows:

1. Run the editor with administrator rights.
2. Open the *C:\inetpub\wwwroot\am\uc\Web.config* configuration file.
3. To specify the URL for connecting to Core Server, for the `Url` parameter in the `amAuthServer` tag set the value, such as *http(s)://full_dns_name of the server/am/core/*.

```
<amAuthServer Url="https://amserv.axidian-id.local/am/core/"/>
```

> ⓘ **NOTE**
> To ignore server certificate errors, in the *am\uc\Config\ApplicationSettings.config* file, change the `isIgnoreCertErrors` parameter to `true`.

4. Start IIS Manager, select **Default Web Site** → **am** and open the **uc** application.

5. Open **Authentication** and enable the following methods:
    - **ASP.NET**
    - **Windows Authentication**
6. Disable all other methods.
7. Open *C:\inetpub\wwwroot\am\uc\Web.config\Web.config* for editing.
8. For the `amAuthentication` tag, in `the mode` parameter, specify `Windows`. Optionally, you can add the `enableLogout="true` parameter to configure the ability to log out of the User Console.

> ⓘ **NOTE**
>
> Leave the `loginUrl` parameter unchanged. When using Windows authentication, this parameter is not taken into account.

```
<amAuthentication mode="Windows" loginUrl="" identityProviderCertificateThumbprint=""
serviceProviderCertificateThumbprint="" enableLogout="true"/>
```

9. Save the changes and restart the IIS server.

The User Console will be available at: *http(s)://full_dns_name of the server/am/uc/*.

Identity Provider

Authentication using Identity Provider

You can set up a secure login to User Console using Identity Provider.

> ⚠ **CAUTION**
>
> To log in to User Console using Identity Provider, you need to install and configure an authentication provider.

Edit the User Console configuration file

> ⚠ **CAUTION**
>
> Before editing the configuration file, follow these steps:
>
> 1. Start IIS Service Manager.
> 2. Select <server Name> → **Sites** → **Default Web Site** → **am** → **uc**.
> 3. Select **Authentication** and enable the options **Anonymous Authentication** and **Authentication using forms**.

1. Open *C:\inetpub\wwwroot\am\uc\web.config*.
2. In the `amAuthServer` Url parameter, specify the address of Core Server, such as *http(s)://full_dns_name of the server/am/core/*.

```
<amAuthServer Url="https://amserv.axidian-id.local/am/core/"/>
```

3. In the `amAuthentication` parameter, do the following:
   - In the mode line, specify the `Saml` value.
   - In the `loginUrl` line, specify the Identity Provider address, such as *http(s) format://full_dns_name of the server/am/idp/*.

```
<amAuthentication mode="Saml" loginUrl="https://server.test.local/am/idp/>
```

   - In the `identityProviderCertificateThumbprint` line, specify the fingerprint of the Identity Provider certificate.

> 💡 **TIP**
>
> If Identity Provider and User Console are installed on different servers, export the certificate without the private key from the server with the Identity Provider module to the server with User Console. The certificate must be added to the certificate store in *Local Machine → Personal*.

```
Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=idp"}
```

- In the `serviceProviderCertificateThumbprint` line, specify the fingerprint of the User Console certificate that was generated during the installing.

  > **INFORMATION**
  > The certificate is installed in the *LocalMachine → Personal* certificate store with the common name `ucsp`.

  ```
  Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=ucsp"}
  ```

4. Save the changes and restart the IIS server.

Edit the Identity Provider configuration file

> **CAUTION**
> Before editing the configuration file, follow these steps:
>
> 1. Start IIS Service Manager.
> 2. Select <server name> → **Sites** → **Default WebSite** → **am** → **idp**.
> 3. Select **Authentication and disable the Windows Authentication** option.

1. Open *C:\inetpub\wwwroot\am\idp\app-settings.json* for editing.

2. In the `PartnerServiceProviderConfigurations` parameter, perform the following actions:

   - In the `SingleLogoutServiceUrl` line, specify the address of the server where User Console was deployed, such as *http(s)://full_dns_name of the server/am/uc/Account/SLOService*.

   - In the `Thumbprint` line, specify the fingerprint of the User Console certificate.

     > **TIP**
     > If Identity Provider and User Console are installed on different servers, export the certificate without the private key from the server with the deployed User Console to the server with the Identity Provider module. The certificate must be added to the certificate store in *LocalMachine → Personal*.

     ```
     Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=ucsp"}
     ```

```
"PartnerServiceProviderConfigurations":
          {
            "Name": "urn:axidian:selfservice",
            "SingleLogoutServiceUrl":
"https://server.axidian.local/am/uc/Account/SLOService",
            "PartnerCertificates": [
              {
                "Thumbprint": "C77EDF29EA05B468BDAF553DE3D804DA4B139C1E"
```

3. In the `AllowWinPwdIfAuthMethodsAreNotAvailable` parameter, you can set the value to `true`, which will allow you to log in to User Console using Windows Password via IDP, even if this method is not specified in the *app-settings.json* configuration file. If the `AllowWinPwdIfAuthMethodsAreNotAvailable` parameter is set to `true`, the following scenarios are possible:
    - Only the registered and user-accessible authentication methods are displayed. With multi-factor authentication, if at least one of the authentication methods is not available, the entire chain is not displayed.
    - If there are no authentication methods available, including methods that do not require registration in User Console (Windows Password, Email, and SMS), Windows Password is displayed.
4. Save the changes and restart the IIS server.

Configure logout from User Console using Identity Provider

1. Open *C:\inetpub\wwwroot\am\uc\Web.config* for editing.
2. In the `amAuthentication parameter`, add the `enableLogout` parameter with the value `true` (the default value is `false`).

```
<amAuthentication mode="Saml" loginUrl="https://server.test.local/am/idp/"
enableLogout="true"/>
```

3. Open *C:\inetpub\wwwroot\am\idp\app-settings.json* for editing.
4. Change the `PartnerServiceProviderConfigurations` parameter. In the `stringSingleLogoutServiceUrl`, specify the server address for User Console, such as *http(s)://full_dns_name of the server/am/uc/Account/SLOService*.

```
"PartnerServiceProviderConfigurations": [
          {
            ...
            "SingleLogoutServiceUrl":
"https://server01.test.local/am/uc/Account/SLOService",
            ...
            ]
          }
```

5. Save the changes and restart IIS.

Example of logging in to User Console using Identity Provider

1. Open User Console in the browser at *http(s)://full_dns_name of the server/am/uc/*.
2. In the authentication window that appears, click **Back** to select the authentication method. By default, the last method used is suggested.
3. To select the authentication method, click **Select**.

> **INFORMATION**
> If a user does not have a registered authenticator, select **Windows Password**.

> **NOTE**
> Logging out of Identity Provider in the user session does not lead the user exiting User Console until the browser restarts or the cookie retention period expires. Cookie retention time for Identity Provider is 30 minutes.

4. Enter the password and click **Sing in**. If the data is entered successfully, the user profile appears.
5. To exit User Console, perform the following steps:

> **NOTE**
> This option is active if you have configured the `enableLogout` parameter.

- Click the username at the top of the window.
- From the drop-down list, select **Exit**.

> **INFORMATION**
> When exiting User Console, you automatically exit Identity Provider.

Enable brute-force protection

In Axidian Access, you can configure account selection protection for the User Console component.

If this setting is enabled, then when you enter a non-existent username, Axidian Access simulates the login of an existing user: displays authentication methods, asks for a password, and then displays the error *Invalid username or authenticator, or the device is locked*.

If this setting is disabled, then when a non-existent username is entered, Axidian Access displays the error *Internal Server error: User not found*.

This setting is disabled by default.

To enable brute-force protection, follow these steps:

1. Open *C:\inetpub\wwwroot\am\core\Web.config*.
2. In the `appSettings` parameter, in the `Bruteforceprotectionapps` line, specify the `Self Service` and `Identity Provider` values.

```
<appSettings>
    ...
    <add key="bruteForceProtectionApps" value="Self Service, Identity Provider" />
</appSettings>
```

3. Save the changes and restart the IIS server.

Set the session lifetime

> **INFORMATION**
> This setting is optional.

To change the session lifetime in User Console, perform the following steps:

1. Open *C:\inetpub\wwwroot\am\uc\Config\ApplicationSettings.config*.

2. Set the required value for the `sessionExpirationTimeInMinutes` parameter. The default value is 30 (minutes).

```
<amApplicationSettings isIgnoreCertErrors="false" sessionExpirationTimeInMinutes="30"/>
```

3. Save the changes.

Display/hide version number in User Console

You can remove User Console headers with the information about the platform and version number, perform the following actions:

1. Open *C:\inetpub\wwwroot\am\uc\Web.config*.
2. Depending on the version of IIS:
   - For IIS 10: Remove comments from the following lines:

     ```
     <security>
           <requestFiltering removeServerHeader="true" />
        </security>
     ```

   - For IIS 10 and above, perform the following actions:
     - Set Rewrite URL.
     - Add the following lines to the `<system.webServer>` section:

       ```
       <rewrite>
       <outboundRules>
           <rule name="replace server header" patternSyntax="Wildcard" lockItem="false">
             <match serverVariable="RESPONSE_SERVER" pattern="*" />
             <action type="Rewrite" value="MyServer" />
           </rule>
       </outboundRules>
       </rewrite>
       ```

Disable localization changes

To prevent a user from changing the language in User Console:

1. Open *C:\inetpub\wwwroot\am\uc\Web.config*.
2. For the `IsEnabled` parameter of the `amCulture` tag, set the `false` value.

```
<amCulture isEnabled="false" />
```

3. Restart the IIS server.

Key Server

Axidian Key Server (Key Server) is the server required for the operation of the Axidian Key mobile application. Push notifications are sent using Key Server and communication with Core Server is established.

Communication with the Axidian Key mobile application is carried out through the HTTP or HTTPS protocol.

When using the SSL connection, you need a root certificate and a web server authentication certificate. When creating a certificate for Key Server, in the Common Name, specify the name or IP of the server to which access for the Axidian Key application is configured in the `ServerUrl` tag of the Key Server configuration file. If the IP address is specified in CN, then it must be without a protocol and port.

To record server events, you need a separate Microsoft SQL or PostgreSQL database.

Installation files

The files for the Key Server installation are located at *Axidian Access <version number>/Axidian Key Server/<version number>*:

- *AxidianKey.Server-<version number>.x64.en-us.msi*: Package for installing Axidian Key Server.
- *AxidianKey.EventLog.Setup-<version number>.x64.en-us.msi*: Package for creating the required log structure in Windows EventLog.

Create a database and a service account

Key Server supports PostgreSQL and Microsoft SQL databases.

For more information about creating a database and a service account, see Data storage.

> **NOTE**
> After you create the database (after the first query), you can reduce the rights for a user. The *db_owner* right is sufficient for the created database.

HTTPS binding in IIS Manager

Key Server is a web application that runs on the AIS database. During installation, SSL is required for Key Server in the settings, which in turn requires HTTPS connection.

1. In IIS Services Manager, in the left menu, navigate to **Axidian Access** → **sites** → **Default Web Site**.
2. In the **Actions** section of the right menu, click **Bindings**.
3. Click **Add** and set the following parameters in the window that appears:
   - In the **Type** field, select *https*.
   - In the **Port** field, specify *443*.
   - In the **SSL Certificate** field, select the Core Server certificate.
4. Click **OK** to save the link.

If you do not plan to use the HTTPS protocol, disable the SSL certificate requirement in the IIS settings for Key Server.

Install and configure Key Server

> **INFO**
> To work correctly, Key Server requires ASP.NET 4.7.

System requirements

1. To install Key Server, run the *AxidianKey.Server-<version number>.x64.en-us.msi* package.
2. Open IIS Manager.
3. Select *Axidian Key Web Site*. In the **Actions** window, select **Bindings** and click **Edit**.
4. In the **Change Site Binding** window, select the SSL certificate with the **Server Authentication** value and click **Ok**.

> **TIP**
> To save the changes in the application configuration file, run the editor with administrator rights.

5. If you use the HTTP protocol, add a binding with a random port and click **Ok**. Open *C:\inetpub\wwwroot\axidiankey\Web.config* and disable the HTTPS requirement as follows: in the `requireHttps` key, change the value from `true` to `false`.

Example

```
<add key="requireHttps" value="false" />
```

6. In the `rootDbContextId` parameter, set the `mssql` value if you use the Microsoft SQL database, or `PostgreSQL` if you use the PostgreSQL database.

7. In the `dbContextSettings` section, specify the string to connect to the database:

   - Microsoft SQL: Use the `mssqlDbContexts` parameter.
   - PostgreSQL: Use the `postgresqlDbContexts` parameter.

> **NOTE**
> Windows authentication is not supported in this version.

8. In the `value` parameter with the `serverUrl` key, specify the URL to connect to Key Server. By default, a secure connection is used (the default port is `81`). When using the HTTP connection, specify a port other than 80. Port `80` is used as the default port for HTTP connections on the IIS server.

> **INFO**
> Key Server must be accessible from a mobile device.

   - For HTTPS: `<add key="serverUrl" value="https://axidiankey.axidian.local:81 " />`
   - For HTTP: `<add key="serverUrl" value="http://axidiankey.axidian.local:82 " />`

9. In the `Url` parameter of the `LogServer` tag, specify the URL to connect to Log Server.

```
<logServer Url="http://logserver.axidian.local/ls/api" CertificateThumbprint=""
CertificateFilePath="" CertificateFilePassword="" />
```

10. Save the changes and restart the IIS server.

Configure server events

Install and configure server events on Log Server.

You can use all the storage methods supported in Log Server.

You can track events by the following ways:

- When stored in Windows events: Via Windows Events.
- When stored in a database: By third-party monitoring tools or using SQL queries.
- When using Syslog: Third-party monitoring tools with Syslog support.

> ⓘ **INFO**
> Current version of Axidian Access does not support viewing Key Server events using Management Console.

To configure Key Server events, perform the following steps:

1. Install *Log Server.AirKey.EventLog.Setup*.
2. Open *C:\inetpub\wwwroot\ls\clientApps.config*.
3. For the `SchemaId` parameter with the `akcSchema` value, set the storage location of the log files. In the `ReadTargetId` tag and the nested `TargetID` tag of the `WriteTargets` tag, set the required parameter:
    - `akcEventLogTarget`: Events are read and written from a separate Windows event log. This value is set by default.

      To view events, open the Windows Log Server event Log and navigate to **Control Panel** → **Administration** → **Event View** → **Application and Service Logs** → **Axidian Key** → **Operational**.
    - `akcSqlTarget`: Events are read and written from Microsoft SQL databases.

      To view events, use the Axidian Access API. Create the SQL query in the EventEntities table.

Example

```
<Application Id="akc" SchemaId="akcSchema">
        <ReadTargetId>akcSqlTarget</ReadTargetId>
   <WriteTargets>
        <TargetId>akcSqlTarget</TargetId>
   </WriteTargets>
   <AccessControl> <!--<CertificateAccessControl CertificateThumbprint="001122...AA11"
Rights="Read" />--> </AccessControl> </Application>
```

If you use a database as the event storage, configure the connection to the database in the *C:\inetpub\wwwroot\ls\targetConfigs\akcSqlTarget.config*.

> **INFO**
>
> For Key Server events, use a separate database.

Remove obsolete data

In the *C:\inetpub\wwwroot\axidiankey\Web.config* file, you can set the parameters of the service for removing old data from the `AkDatas` table.

```
<jobSettings>
      <cleanAkDatas enabled="true" firstRunTime="00:00" intervalSec="86400"
akDataLifeTimeSec="86400" batchDeleteDelayMsec="1000" />
</jobSettings>
```

Description of attributes from the example

Attribute name	Description	Value	Default value	The minimum allowed value
`enabled`	The flag for enabling the service	`true` \ `false`	`true`	Absent
`firstRunTime`	The time of the first launch of the service	A string in the `hh:mm` time format	`00:00` (midnight)	Absent
`intervalSec`	Service launch interval	Number of seconds	`86400` (24 hours)	`600`

Attribute name	Description	Value	Default value	The minimum allowed value
`akDataLifeTimeSec`	Data lifetime	Number of seconds	`86400` (24 hours)	`600`
`batchDeleteDelayMsec`	Interval between data deletions	Number of milliseconds	`1000`	`0`

Healthcheck

To check the status of Key Server, use the `api/HealthCheck` method. The server checks the storage status in the background task. The interval for executing the background task is configured in the *C:\inetpub\wwwroot\axidiankey\Web.config* file, the `jobSettings` section:

Example

```
<jobSettings>
      <storageHealthCheck intervalSec="60" />
  </jobSettings>
```

Encrypt/Decrypt the configuration file

Use the following batch files located in the Axidian Access distribution at *Axidian version number folder\Axidian Key Server\8.2.2\Misc\EA.Config.Encryptor*:

- *EncryptConfig.bat*: Encrypt the configuration file.
- *DecrypConfig.bat*: Decrypt the configuration file.

Reset Password

Axidian Reset Password (Reset Password) is a utility designed to reset a user's password with automatic synchronization in the Axidian Access system.

When resetting the password through this utility, the password will be generated automatically. The complexity of the generated password is set by the system administrator when configuring the utility. The generated password is sent to the user by email or SMS.

The utility resets the user password in Active Directory, so for a user on whose behalf the utility is running, you must grant administrator rights in the domain. It is enough to add a user to the *Administrators* domain group.

The user also must have global administrator rights in the Axidian Access system.

> ⚠ **CAUTION**
> To send a password to a user by email or SMS, you must install providers SMS OTP or Email OTP on Core Server.

Installation files

The files for installation of the utility are located at *Axidian Access version number\Axidian Reset Password*.

- *Axidian.EA.ResetPassword version number.x64.en-us.msi*: Package for installing the Reset Password utility.
- *\Misc*: Folder with group policy templates.

Install Reset Password

> ⓘ **NOTE**
> Install the utility on the workstation/server, where the utility will be run in the future.

To install the utility, run *Axidian.EA.ResetPassword<version number>.x64.en-us.msi*.

Configuration

1. Download the Axidian Access GPO templates from the distribution to the local storage of the computer where the utility is installed.

   > ⓘ **INFO**
   > The local GPO storage is located at *C:\Windows\PolicyDefinitions*.

2. Open Local Group Policy Editor and go to **Computer Configuration** → **Administrative Templates** → **Axidian Access** → **Reset Password**.

3. Select and configure the *Password Reset Utility Setup* policy. Set the following values:

   - *Server URL*: URL of Core Server, such as *http(s)://DNS name of the server/am/core/*.
   - *Provider type*: The method of delivering the generated password to the user. You can use SMS or email.
   - *Message subject text*: The subject of the message with the generated password.

   > ⓘ **INFO**
   > This is a required parameter when using email to receive a password.

   - *Message text*: Text of the message with the generated password.

   > ⓘ **INFO**
   > Optional parameter.
   >
   > If the value is not set, the generated password itself is used as the message text.
   >
   > If a value is specified, the field must contain the text *<password>*.

4. Select and configure the *Password Generation* configuration policy. In the policy, specify the required criteria for password generation.

Example of the utility

1. From the Windows Start menu, run the Reset Password utility or execute *C:\Program Files\Axidian\Reset Password\EA.ResetPassword.exe* on behalf of the user who has the rights to reset the password in Active Directory.

2. In the main window of the utility, click **Login**.

> **NOTE**
>
> Logging in to the utility is performed under the user in whose session the startup is performed. If required, you can run the utility on behalf of another user in an active session.

3. Search for the user who needs to reset the password, and select it in the search results.

4. On the user profile, click **Reset AD password** and confirm the action.

> **NOTE**
>
> When resetting the password, pay attention to the **Phone Number** and **Email** fields. To send the password by SMS, the user's phone number must be specified in the **Phone Number** field. To send it by email, the email address must be specified in the **Email** field.

5. If the password has been successfully reset, a message appears stating that the password has been successfully reset and the message has been sent.

Location of configuration files

After you install Axidian Access, the configuration files of the components are created automatically. By default, they are stored in the *C:\inetpub\wwwroot* directory.

The table below lists the full paths to the configuration files.

Component	Configuration file
Log Server main configuration file	*C:\inetpub\wwwroot\ls\clientApps.config*
Core Server	*C:\inetpub\wwwroot\am\core\Web.config*
Management Console	*C:\inetpub\wwwroot\am\mc\Web.config*
User Console	*C:\inetpub\wwwroot\am\uc\Web.config*
Key Server (Axidian Key) 8.1.x and lower	*C:\inetpub\wwwroot\airkeycloud\Web.config*
Key Server (Axidian Key) 8.2 and higher	*C:\inetpub\wwwroot\axidiankey\Web.config*
Identity Provider (IDP) 8.1.4 and lower	*C:\inetpub\wwwroot\am\idp\Web.config*
Identity Provider (IDP) 8.1.5 and higher	*C:\inetpub\wwwroot\am\idp\app-settings.json*
Event storage configuration in Microsoft SQL	*C:\inetpub\wwwroot\ls\targetConfigs\sampleDb.config*
Event storage configuration in PostgreSQL	*C:\inetpub\wwwroot\ls\targetConfigs\postgresDb.config*
Event storage configuration in SysLog	*C:\inetpub\wwwroot\ls\targetConfigs\sampleSyslog.config*

NOTE

- For Identity Provider 8.1.5 and higher, the *app-settings.json* configuration file is used instead of *Web.config*.
- In Axidian Access 8.2 and higher, the Key Server component is renamed to Axidian Key. Accordingly, the directory is renamed to *axidiankey* instead of *airkeycloud*.

Install integration modules

Identity Provider

5 items

ADFS Extension

2 items

IIS Extension

4 items

RDP Windows Logon

3 items

NPS RADIUS Extension

3 items

Mobile Device Provisioning

Configuring the synchronization of a device placed in quarantine

Enterprise Single Sign-On

6 items

Windows Logon

6 items

Authenticator management utility

4 items

Identity Provider

The Identity Provider module (formerly SAML IDP) is used to organize multifactor authentication and single sign-on to web applications.

Starting from Axidian Access 8.1.5, the Identity Provider module supports the following protocols:

- OpenID Connect and OAuth 2.0
- SAML

> ⚠ **IMPORTANT**
> If you use Axidian Access 8.1.4 or lower, you can find the required information about installing and configuring the module in the Axidian Access 8.1 documentation.

Identity Provider relieves the user of the need to memorize multiple credentials: only one set of credentials is required to access all integrated systems. Authentication is performed centrally on the Identity Provider side.

In Identity Provider, you can authenticate with the following authenticators:

- Email OTP
- Hardware OTP
- Hardware TOTP
- Passcode
- Secured TOTP
- SMS OTP
- Software OTP
- Storage SMS OTP
- Telegram Provider (in the mode of sending one-time passwords and push notifications with login confirmation)
- Windows Password
- the Axidian Key mobile application (in the mode of sending one-time passwords and push notifications with login confirmation)

Prerequisites

Before installing Identity Provider:

1. Install .NET 6 on the server.

   To do this, run the `dotnet-hosting-6.0.10-win.exe` utility located at `Axidian Access <version number>/Axidian Idp`.

   Correct operation of Identity Provider is guaranteed only when .NET 6 is used.

2. Install the required Internet Information Services components.

   To do this, run the `Axidian.SAML.IIS.Install.MSServer.ps1` script from `Axidian Access <version number>/Misc/IISScripts` in PowerShell as an administrator.

   If the security settings do not allow running the script, open `Axidian.SAML.IIS.Install.MSServer.ps1` in Notepad, copy all the commands, and run them in PowerShell as an administrator.

> ⚠ **IMPORTANT**
> By default, Identity Provider uses Windows authentication. For non-domain scenarios, enable anonymous authentication.

<details>
<summary>Enable anonymous authentication</summary>

1. In IIS Manager, in the left menu, select **Axidian Access** → **sites** → **Default Web Site** → **am** → **idp** and click **Authentication**.
2. Enable the **Anonymous Authentication** option and leave the other options disabled.

</details>

Install and configure Identity Provider

To install and configure Identity Provider:

1. Run `Axidian.IDP-<version number>.en-us.msi` located at `Axidian Access <version number>/Axidian Idp`.

By default, a new IDP server certificate is created and installed after the package is installed. The certificate encrypts the data transmitted between Core Server and the end applications, such as Management Console.

The certificate is not intended for establishing an SSL connection.

If the certificate was generated earlier, clear the **Create and install a new IDP server certificate** check box in the installation wizard window.

2. Add a binding in the IIS settings:

    1. In IIS Manager, in the left menu, select **Axidian Access** → **sites** → **Default Web Site**.
    2. Click **Bindings** in the **Actions** section of the right menu.
    3. Click **Add** and specify the following parameters in the window that opens:
        - In the **Type** field, select **https**.
        - In the **Port** field, specify 443.
        - In the **SSL certificate** field, select the web server certificate.
    4. Click **OK** to save the binding.
    5. (Optional) If you use Windows Server 2022 or higher, enable the **Disable TLS 1.3 over TCP** option and then restart IIS.

NOTE

Identity Provider is a web application that runs on IIS. For security reasons, Identity Provider works only over the HTTPS protocol. Do not change the protocol, as this reduces the connection security.

If you plan to use the HTTP protocol, edit the `Web.config` file in the `C:\inetpub\wwwroot\am\idp` folder.

Modify the *Web.config* file

In the `aspNetCore` tag, add the following lines:

```
<environmentVariables>
 <environmentVariable name="ASPNETCORE_ENVIRONMENT" value="Development" />
</environmentVariables>
```

Result:

```
<aspNetCore processPath="dotnet" arguments=".\AM.IDP.dll" stdoutLogEnabled="false"
stdoutLogFile=".\logs\stdout" hostingModel="inprocess">
<environmentVariables>
 <environmentVariable name="ASPNETCORE_ENVIRONMENT" value="Development" />
</environmentVariables>
</aspNetCore>
```

Modify the configuration file

To configure the Identity Provider module:

1. Open the `app-settings.json` configuration file located at `C:\inetpub\wwwroot\am\idp`.
2. In the `AuthenticationServer` section, in the `Url` parameter, specify the Core Server address in the format `http(s)://full_dns_server_name/am/core/`.

```
"AuthenticationServer": {
  //highlight-grey-next-line
    "Url": "AUTH_SERVER_URL",
    "IsIgnoreCertErrors": false
  }
```

Server address example

```
https://server.test.local/am/core/
```

> ⚠ **IMPORTANT**
>
> For correct operation, we recommend that you install a client certificate on each Core Server.
>
> To ignore server certificate errors, change the value of the `IsIgnoreCertErrors` parameter to *true*.

3. In the `Server` section, in the `Url` parameter, specify the address for connecting to Log Server in the format `http(s)://full_dns_server_name/ls/api`.

```
"Server": {
      "Url": "LOG_SERVER_URL"
}
```

Log Server address example

```
https://server.test.local/ls/api/
```

4. In the `LocalIdentityProviderConfiguration` section, set the values of the following parameters:
   - In the `SingleSignOnServiceUrl` parameter, specify the Identity Provider address in the format `http(s)://full_dns_server_name/am/idp/Account/SsoService`.

```
"LocalIdentityProviderConfiguration": {
          "Name": "urn:axidianid:saml_idp",
          //highlight-grey-next-line
          "SingleSignOnServiceUrl": "IDENTITY_PROVIDER_URL/Account/SsoService"
```

SSO server address example

```
https://server.test.local/am/idp/Account/SsoService
```

- In the `Thumbprint` parameter, specify the thumbprint of the Identity Provider certificate that was generated during the installation.

```
"LocalIdentityProviderConfiguration": {
          "LocalCertificates": [
            {
            //highlight-grey-next-line
              "Thumbprint": "YOUR_CERTIFICATE_THUMBPRINT"
            }
          ]
        }
```

To get the certificate thumbprint, use the following PowerShell request:

```
Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=idp"}
```

5. In the `AuthenticationMethods` section, delete the unnecessary lines and add the identifiers of the providers you plan to use in the following format:

   - In the `Name` parameter, specify an arbitrary unique value.
   - In the `Providers` parameter, specify the identifier of the provider used.

Identifiers available for Identity Provider

- SMS OTP {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}
- Storage SMS OTP {3F2C1156-B5AF-4643-BFCB-9816012F3F34}
- Email OTP {093F612B-727E-44E7-9C95-095F07CBB94B}
- Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
- Software OTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
- Secured TOTP {F15FD7EC-19EA-4384-846E-A2D0BE149FA2}
- Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
- Hardware TOTP {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
- Axidian Key {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}
- Windows Password {CF189AF5-01C5-469D-A859-A8F2F41ED153}
- Telegram Provider {CA4645CC-5896-485E-A6CA-011FCC20DF1D}

Example of using one provider

```
"AuthenticationMethods": [
      {
        "Name": "Passcode",
        "Providers": [
          "F696F05D-5466-42b4-BF52-21BEE1CB9529"
        ]
      }
```

Example of using several providers

```
"AuthenticationMethods": [
     {
        "Name": "HOTP_Passcode",
        "Providers": [
          "AD3FBA95-AE99-4773-93A3-6530A29C7556",
          "F696F05D-5466-42b4-BF52-21BEE1CB9529"
        ]
      }
```

6. In the `SelfService` section, set the `AllowWinPwdIfAuthMethodsAreNotAvailable` parameter to one of the following values:

   - If you set the value to `true`, users can log in to User Console through Identity Provider using Windows Password even if they have no authenticator. This value also allows logging in to User Console with any authenticator, even if it is not specified in the `AuthenticationMethods` section.

   - If you set the value to `false`, authorization with Windows Password is not available if this authentication method is not present in the `AuthenticationMethods` section. Logging in to User Console is possible only with the authenticator specified in this section.

> ⓘ **NOTE**
> If you use authentication both with Windows Password and with a provider, the event log displays the following:
>
> - If Windows Password was entered correctly and the provider incorrectly, a successful login to Identity Provider with Windows Password is registered in the user events.
> - If both Windows Password and the provider were entered correctly, a successful login with the provider is registered in the user events.

7. In the `OIDC` section, in the `CertificateThumbprint` parameter, specify the thumbprint of the Identity Provider certificate that was generated during the installation.

To get the certificate thumbprint, use the following PowerShell request:

```
Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=idp"}
```

8. (Optional) To configure Axidian Access 8.2 to work with Mobile Device Provisioning 9.x and authentication through IDP 8.2, add the following parameters for the `urn:axidianid:mobiledeviceprovisioning` application in the `SAML` block:

   - `SignLogoutResponse` with the value `true` — this parameter is required for the **Logout** button in the MDP 9.x interface to work correctly; without this setting an error may appear.
   - `Thumbprint` with the thumbprint of the Identity Provider 8.2 certificate.

To get the certificate thumbprint, use the following PowerShell request:

```
Get-ChildItem Cert:\LocalMachine\My\ | Where-Object { $_.Subject -match "CN=idp" }
```

Example

```
{
 "Name": "urn:axidianid:mobiledeviceprovisioning",
 "SingleLogoutServiceUrl": "https://amcore.axidian-test.com/am/mdp/Account/SLOService",
 //highlight-grey-next-line
 "SignLogoutResponse": true,
 "PartnerCertificates": [
    {
    //highlight-grey-next-line
    "Thumbprint": "67446E7995669269987637CA26F953D4B9E9F2DC"
    }
 ]
},
```

9. Save the configuration file and restart IIS.

Integration with applications over OpenID Connect and OAuth 2.0

To integrate Identity Provider with applications over the OpenID Connect and OAuth 2.0 protocols, use the following parameters of the `am/idp/app-settings.json` configuration file.

To get the server metadata, follow the link: `https://<server_dns>/am/idp/.well-known/openid-configuration`.

Parameter	Description
`CustomAttributes`	The block specifies the `ServiceProvider` and `Attributes` attributes (with the `Name` and `UserNameFormat` parameters).
`ServiceProvider`	The parameter value corresponds to `ClientId` from the `OIDC` section.
`Name`	Defines the key the transmitted attribute is displayed with as a result of decrypting the IdToken value or a request to Userinfo. If the `Name` parameter does not match one of the possible values (`email`, `name`, `family_name`, `given_name`, `middle_name`), the transmitted attribute is not displayed in the response to the Userinfo request, but it is displayed in the IdToken decryption.
`UserNameFormat`	Defines the user information transmitted to the OIDC client. This parameter accepts a strictly defined list of values: Id, ObjectGUID, Name, CanonicalName, PrincipalName, SamCompatibleName, DistinguishedName, Sid, FirstName, MiddleName, LastName, Email, Phone.
`CertificateThumbprint`	The thumbprint of the Identity Provider certificate. The certificate is loaded into the store of the computer where Identity Provider is installed.
`Clients`	The block specifies the settings for each client application. There can be several client applications.
`ClientId`	A unique identifier used to define the client application when exchanging tokens, as well as for user authentication and authorization. When a client application requests access to protected resources, it provides its `ClientId` along with other credentials to obtain an access token. Default value: *example-client*.

Parameter	Description
`ClientSecret`	A string of characters known only to the client application and the IDP server. `ClientSecret` is used in the token exchange process to confirm the identity of the client application. When requesting an access token, the client must provide its `ClientId` and `ClientSecret` for authentication. Default value: *secret_secret_secret*.
`DisplayName`	The name of the client application. Used to display information in the client interface. Default value: *Example client application*.
`Permissions`	The block specifies the attributes allowed for the application. After successful authentication and granting of permissions, the application can use the obtained data. The data specified in `Permissions` can be transmitted in the response over the OIDC protocol (ept — Endpoints, gt — GrantTypes, rst — ResponseTypes, scp — Scopes).
`ept:authorization`	Initiating authorization.
`ept:logout`	Terminating the session associated with the token by identifier.
`ept:token`	Used to obtain a token.
`gt:authorization_code`	Used to obtain tokens (ID Token and Access Token) with an intermediate authorization code.
`gt:refresh_token`	Used to obtain a new access token without re-authenticating the user.
`rst:code`	Used to obtain an authorization code.
`scp:email`	The user email address.
`scp:profile`	The user profile information includes the following components: `name`, `given_name`, `family_name`, `middle_name`.
`scp:openid`	Indicates that the client application requests user authentication.
`scp:offline_access`	Allows the client application to request a refresh token.
`PostLogoutRedirectUris`	Contains the valid URLs the user is redirected to after logging out of the client application.

Parameter	Description
`RedirectUris`	Contains the valid URLs the user is redirected to after a successful login to the client application.
`Requirements`	Defines additional requirements for client requests that must be met for successful user authentication and authorization.

The following table lists the attributes that Identity Provider transmits to the client application after successful user authentication.

Parameter	Description
`exp`	Defines the time after which the ID Token is not accepted.
`iat`	The time the JWT was issued.
`sub`	A unique subject identifier. The user identifier is specified as the value.
`iis`	The organization that issued the token. A URL is specified.
`aud`	The token recipient; the `client_id` of the application that sent the authentication request is specified.

For more information about configuring integration with applications over the OpenID Connect and OAuth 2.0 protocols, see the knowledge base articles Configuring OIDC using Keycloak as an example and Configuring NextCloud + OIDC.

Integration with applications over SAML

- Identity Provider Login URL — `http(s)://<full_DNS_name_of_the_Identity_Provider_server>/am/idp/Account/SsoService`.
- Identity Provider Logout URL — `http(s)://<full_DNS_name_of_the_Identity_Provider_server>/am/idp/Account/Logout`.
- Identity Provider Name — *urn:axidianid:saml_idp*.

To integrate Identity Provider with applications over the SAML protocol, use the following parameters of the `am/idp/app-settings.json` configuration file.

Parameter	Description
`SingleSignOnServiceUrl`	Contains the URL of the Single Sign-On Service that receives SAML messages about logging in to the client application.
`LocalCertificates`	The Identity Provider certificate. The certificate is loaded into the store of the computer where Identity Provider is deployed. The certificate is specified in the `Thumbprint` parameter.
`SingleLogoutServiceUrl`	Contains the URL of the Single Logout Service that receives SAML messages about logging out of the client application.
`PartnerCertificates`	The client application certificate. The certificate is specified in the `Thumbprint` parameter.
`Name`	The name of the client application.
`Description`	The description of the client application.
`WantAuthnRequestSigned`	Defines whether the SAML authentication request must be signed. Recommended value: *true*.
`SignSAMLResponse`	Defines whether the SAML responses sent to the client application must be signed. Recommended value: *true*.
`WantSamlResponseSigned`	Defines whether the SAML assertions must be encrypted. Recommended value: *true*.

Parameter	Description
`SignLogoutRequest`	Defines whether the SAML logout requests sent to the client application must be signed. Recommended value: *true*.
`SignLogoutResponse`	Defines whether the SAML logout responses sent to the client application must be signed. Recommended value: *true*.
`SignAssertion`	Defines whether the SAML assertions must be signed. Recommended value: *true*.
`WantLogoutRequestSigned`	Defines whether the SAML logout requests received from the client application must be signed. Recommended value: *true*.
`WantLogoutResponseSigned`	Defines whether the SAML logout responses received from the client application must be signed. Recommended value: *true*.
`AssertionConsumerServiceUrl`	Contains the URL of the Assertion Consumer Service of the client application that receives SAML requests.

For an example of integration over the SAML protocol between IDP and Nextcloud, see the knowledge base.

Enable brute-force protection

In Axidian Access, you can configure protection against account brute-forcing for the Identity Provider component.

If this setting is enabled, when a non-existent username is entered, Axidian Access simulates the login of an existing user through Identity Provider: it displays the authentication methods, requests a password, and then displays the error *Invalid username or authenticator, or the device is blocked*.

If this setting is disabled, when a non-existent username is entered, Axidian Access displays the error *Internal server error: User not found*.

By default, this setting is disabled.

To enable brute-force protection:

1. Open the Core Server `web.config` configuration file located in the `C:\inetpub\wwwroot\am\core` folder.

2. Modify the `appSettings` parameter. In the `bruteForceProtectionApps` line, specify the value *Identity Provider*.

   Example

   ```
   <appSettings>
       <add key="requireHttps" value="true" />
       <add key="enableSwagger" value="false" />
       //highlight-grey-next-line
       <add key="bruteForceProtectionApps" value="Identity Provider" />
   </appSettings>
   ```

3. Save the changes and restart IIS.

Authentication by username without specifying the domain

You can configure the domain selection from a drop-down list.

If you specify several domains, the domain listed first is selected by default.

To configure this, perform the following actions:

1. Open the `app-settings.json` configuration file located in the `C:\inetpub\wwwroot\am\idp` folder.
2. Modify the `AuthenticationSettings` parameter. In the `Domains` line, specify the values of the required domains.

   Example

   ```
   "AuthenticationSettings": {
     "SessionExpirationTimeInMinutes": 30,
     "Domains": [
     //highlight-grey-start
       "test.local",
       "demo.local",
       "axidian.local"
     //highlight-grey-end
     ]
   ```

3. Save the file and restart IIS.

Switch the Telegram Provider operating mode

You can log in to Identity Provider with Telegram Provider either with a one-time password or with a push notification requesting to confirm or reject the login.

To select the login confirmation method:

1. In the Management Console sidebar, open the **Policies** section.
2. From the list of policies, select the policy with the Identity Provider application added.
3. Open the **Applications** tab and go to the Identity Provider application.
4. In the **Telegram operating mode** item, select the authentication method: one-time password or push notification.

ADFS Extension

ADFS Extension (2012)

Two-factor authentication using ADFS

ADFS Extension (2016)

Two-factor authentication using ADFS

ADFS Extension (2012)

With the ADFS Extension module, you can implement multifactor authentication for the Microsoft ADFS server, adding a second factor to the access gaining process.

In ADFS Extension, you can authenticate with the following authenticators:

- Email OTP
- Hardware OTP
- Hardware TOTP
- Passcode
- Secured TOTP
- SMS OTP
- Software OTP
- Storage SMS OTP
- Telegram Provider (in the mode of sending one-time passwords and push notifications with login confirmation)
- the Axidian Key mobile application (in the mode of sending one-time passwords and push notifications with login confirmation)

Examples of extension deployment:

- Configuring two-factor authentication for applications published in WAP
- Configuring two-factor authentication in AD FS for integration with Exchange Server 2016
- Configuring two-factor authentication in Microsoft Office 365 with ADFS Extension

Install and configure ADFS Extension

1. Run the installation file located at `Axidian Access <version number>/Axidian ADFS Extension/<version number>` and follow the steps of the installation wizard.

2. Create a configuration file named `MFAAdapter.json` with the following parameters:

    - `EANetServerURL` — the Core Server address

    - `ModeId` — the identifier of the authentication method used

> **MODEID CAN HAVE THE FOLLOWING VALUES:**
>
> - SMS OTP {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}
> - Storage SMS OTP {3F2C1156-B5AF-4643-BFCB-9816012F3F34}
> - Secured TOTP {F15FD7EC-19EA-4384-846E-A2D0BE149FA2}
> - Email OTP {093F612B-727E-44E7-9C95-095F07CBB94B}
> - Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
> - Software OTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
> - Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
> - Hardware TOTP {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
> - Axidian Key Provider {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}
> - Telegram Provider {CA4645CC-5896-485E-A6CA-011FCC20DF1D}

Example

```
{
 "ServerType":"eaNet",
 "EANetServerURL":"https://YourDomainName/am/core/",
 "ModeId":"{0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}"
}
```

> **NOTE**
>
> When using the HTTPS protocol connection, you must install a client certificate on each Core Server.

3. Run PowerShell as an administrator. To register the adapter, enter the following data:

   - `YourPath\MFAAdapter.json` — specify the full path to the `MFAAdapter.json` configuration file created in step 2.
   - In the `$typeName` variable, in the `Version` parameter, specify the version number of the ADFS Extension used.

> **IMPORTANT**
>
> When registering, modifying, or removing the adapter, restart the ADFS services on each ADFS server.

Example

```
$typeName = "Axidian.ADFS.MFAAdapter.MFAAdapter, Axidian.ADFS.MFAAdapter,
Version=1.0.6.0, Culture=neutral, PublicKeyToken=1ebb0d9282100d91"
Register-AdfsAuthenticationProvider -TypeName $typeName -Name "MFA Adapter" -
ConfigurationFilePath 'YourPath\MFAAdapter.json'
```

- To register several providers, change the provider name in the `Name` parameter.

Example

```
Register-AdfsAuthenticationProvider -TypeName $typeName -Name "MFA Passcode" -
ConfigurationFilePath 'YourPath\MFAAdapter.json'
```

- Change the display name of the provider. For the `Name` parameter, specify the value from the previous step; for the `DisplayName` parameter, specify the name that is displayed during authentication through ADFS.

Example

```
Set-AdfsAuthenticationProviderWebContent -Name "MFA Adapter Passcode" -DisplayName
"Passcode"
```

4. To remove the adapter, run the following command:

```
Unregister-AdfsAuthenticationProvider -Name "MFA Adapter"
```

5. To update the configuration, run the following command:

```
Import-AdfsAuthenticationProviderConfigurationData -Name "MFA Adapter" -FilePath
'YourPath\MFAAdapter.json'
```

Enable multifactor authentication for ADFS

1. Open the ADFS management console.
2. Select **Authentication Policies**.
3. In the **Actions** window, select **Edit Global Multi-factor Authentication...**.
4. Add a user/group and enable the following parameters:

- On the **Multi-factor** tab, in the **Location** item, select **Extranet** and **Intranet**.
- Select the *MFA Adapter* provider.

5. Restart the ADFS service to apply the changes.

Register an authenticator at the first login to ADFS

At the first login to ADFS, the user is prompted to specify a phone number in the following cases:

- The user has no phone number specified in Active Directory.
- The phone number is specified, but the *Use the phone number from Active Directory if the authenticator is not registered* policy is disabled.

After specifying the phone number, the user receives a one-time password to that number, which must be entered in the ADFS form to complete the login to the target application.

Switch the Telegram Provider operating mode

You can log in to ADFS with Telegram Provider either with a one-time password or with a push notification requesting to confirm or reject the login.

To select the login confirmation method, do the following:

1. In Management Console, in the left menu, select **Policies**.
2. From the list of policies, select the policy with the ADFS application added.
3. In the policy menu on the left, select **Applications** and go to the ADFS application.
4. In the **Telegram operating mode** item, select the authentication method: one-time password or push notification.

Example of the module operation on the ADFS idpinitiatedsignon page

The example of the extension operation is shown on the `idpinitiatedsignon.htm` page.

By default, this page is not configured. Configuring this page is optional.

Configuring the test page

1. Select **Relying Party Trusts** and click **Add Relying Party Trust...**.
2. On the **Welcome** tab, select **Claims aware** and click **Start**.
3. On the **Select Data Source** tab, specify the URL of your application and click **Next**.

> ⓘ **INFORMATION**
> The standard ADFS page `idpinitiatedsignon.htm` is used as an example of the extension operation. The metadata address for this page is used.

4. On the **Specify Display Name** tab, enter the name and description for your trust and click **Next**.
5. On the **Choose Access Control Policy** tab, select a suitable policy with an MFA request from the default ones; you can also add arbitrary access control policies.
6. Leave the other parameters as default.
7. Restart the ADFS service to apply the changes.

Module operation

1. Open the ADFS test page: https://YourDomainName/adfs/ls/idpinitiatedsignon.htm.
2. Perform the login.
3. After entering the domain login and password, specify the data for the second authentication factor.
4. After the correct input, the login is performed.

ADFS Extension (2016)

With the ADFS Extension module, you can implement multifactor authentication for the Microsoft ADFS server, adding a second factor to the access gaining process.

In ADFS Extension, you can authenticate with the following authenticators:

- Email OTP
- Hardware OTP
- Hardware TOTP
- Passcode
- Secured TOTP
- SMS OTP
- Software OTP
- Storage SMS OTP
- Telegram Provider (in the mode of sending one-time passwords and push notifications with login confirmation)
- the Axidian Key mobile application (in the mode of sending one-time passwords and push notifications with login confirmation)

Examples of extension deployment

- Configuring two-factor authentication for applications published in WAP
- Configuring two-factor authentication in AD FS for integration with Exchange Server 2016
- Configuring two-factor authentication in Microsoft Office 365 with ADFS Extension

Install and configure ADFS Extension

> ⓘ **NOTE**
> Before running the ADFS Extension installation package, add the Active Directory Federation Services (AD FS) role.

1. Run the installation file located at `Axidian Access <version number>/Axidian ADFS Extension/<version number>` and follow the steps of the installation wizard.

2. Create a configuration file named `MFAAdapter.json` with the following parameters:

- `EANetServerURL` — the Core Server address
- `ModeId` — the identifier of the authentication method used

> **MODEID CAN HAVE THE FOLLOWING VALUES:**
>
> - SMS OTP {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}
> - Storage SMS OTP {3F2C1156-B5AF-4643-BFCB-9816012F3F34}
> - Secured TOTP {F15FD7EC-19EA-4384-846E-A2D0BE149FA2}
> - Email OTP {093F612B-727E-44E7-9C95-095F07CBB94B}
> - Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
> - Software OTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
> - Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
> - Hardware TOTP {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
> - Axidian Key Provider {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}
> - Telegram Provider {CA4645CC-5896-485E-A6CA-011FCC20DF1D}

Example

```
{
 "ServerType":"eaNet",
 "EANetServerURL":"https://YourDomainName/am/core/",
 "ModeId":"{0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}"
}
```

> **NOTE**
>
> When using the HTTPS protocol connection, you must install a client certificate on each Core Server.

3. Run PowerShell as an administrator. To register the adapter, enter the following data:
   - `YourPath\MFAAdapter.json` — specify your full path to the `MFAAdapter.json` configuration file created in step 2.
   - In the `$typeName` variable, in the `Version` parameter, specify the version number of the ADFS Extension used.

> **IMPORTANT**
>
> When registering, modifying, or removing the adapter, restart the ADFS services on each ADFS server.

Example

```
$typeName = "Axidian.ADFS.MFAAdapter.MFAAdapter, Axidian.ADFS.MFAAdapter, Version=1.0.12.0, Culture=neutral, PublicKeyToken=1ebb0d9282100d91"
Register-AdfsAuthenticationProvider -TypeName $typeName -Name "MFA Adapter" -ConfigurationFilePath 'YourPath\MFAAdapter.json'
```

- To register several providers, change the provider name in the `Name` parameter.

  Example

  ```
  Register-AdfsAuthenticationProvider -TypeName $typeName -Name "MFA Passcode" -ConfigurationFilePath 'YourPath\MFAAdapter.json'
  ```

- Change the display name of the provider. For the `Name` parameter, specify the value from the previous step; for the `DisplayName` parameter, specify the name that is displayed during authentication through ADFS.

  Example

  ```
  Set-AdfsAuthenticationProviderWebContent -Name "MFA Adapter Passcode" -DisplayName "Passcode"
  ```

4. To remove the adapter, run the following command:

```
Unregister-AdfsAuthenticationProvider -Name "MFA Adapter"
```

5. To update the configuration, run the following command:

```
Import-AdfsAuthenticationProviderConfigurationData -Name "MFA Adapter" -FilePath 'YourPath\MFAAdapter.json'
```

Enable multifactor authentication for ADFS

1. Open the ADFS management console.
2. Select **Service→Authentication Methods**.
3. In the **Actions** window, select **Edit Multi-factor Authentication Methods...**.

4. On the **Multi-factor** tab, select the previously created provider and click **Apply**.

5. Restart the AD FS service to apply the changes.

Register an authenticator at the first login to ADFS

At the first login to ADFS, the user is prompted to specify a phone number in the following cases:

- The user has no phone number specified in Active Directory.
- The phone number is specified, but the *Use the phone number from Active Directory if the authenticator is not registered* policy is disabled.

After specifying the phone number, the user receives a one-time password to that number, which must be entered in the ADFS form to complete the login to the target application.

Switch the Telegram Provider operating mode

You can log in to ADFS with Telegram Provider either with a one-time password or with a push notification requesting to confirm or reject the login.

To select the login confirmation method, do the following:

1. In Management Console, in the left menu, select **Policies**.
2. From the list of policies, select the policy with the ADFS application added.
3. In the policy menu on the left, select **Applications** and go to the ADFS application.
4. In the **Telegram operating mode** item, select the authentication method: one-time password or push notification.

Example of the module operation on the ADFS idpinitiatedsignon page

The example of the extension operation is shown on the `idpinitiatedsignon.htm` page.

By default, this page is not configured. Configuring this page is optional.

Configuring the test page

1. Select **Relying Party Trusts** and click **Add Relying Party Trust...**.
2. On the **Welcome** tab, select **Claims aware** and click **Start**.
3. On the **Select Data Source** tab, specify the URL of your application and click **Next**.

> **INFORMATION**
> The standard ADFS page `idpinitiatedsignon.htm` is used as an example of the extension operation. The metadata address for this page is used, for example, `https://<full_dns_server_name>/federationmetadata/2007-06/federationmetadata.xml`.

4. On the **Specify Display Name** tab, enter the name and description for your trust and click **Next**.
5. On the **Choose Access Control Policy** tab, select a suitable policy with an MFA request from the default ones; you can also add arbitrary access control policies.
6. Leave the other parameters as default.
7. Restart the ADFS service to apply the changes.

Module operation

By default, the `idpinitiatedsignon.htm` page is disabled in ADFS 2016. To enable it, run PowerShell as an administrator and execute the following command:

```
Set-AdfsProperties -EnableIdpInitiatedSignonPage $True
```

1. Open the ADFS test page: https://YourDomainName/adfs/ls/idpinitiatedsignon.htm.
2. Perform the login.
3. After entering the domain login and password, specify the data for the second authentication factor.
4. After the correct input, the login is performed.

IIS Extension

With IIS Extension, you can configure two-factor authentication in web applications that use Forms Authentication and are deployed on the Microsoft Internet Information Services (IIS) platform.

Two-factor authentication is supported only for applications that use Forms Authentication.

Two-factor authentication is implemented with authentication by the domain password and by a second factor — a one-time password or a push notification in the Axidian Key application.

In IIS Extension, you can authenticate with the following authenticators:

- Email OTP
- Hardware OTP
- Hardware TOTP
- Passcode
- SMS OTP
- Software OTP
- Telegram Provider (in the mode of sending one-time passwords and push notifications with login confirmation)
- the Axidian Key mobile application (in the mode of sending one-time passwords and push notifications with login confirmation)

Examples of IIS Extension deployment:

- Configuring one-factor and two-factor authentication in Remote Desktop Web Access (RDWeb) with IIS Extension
- Configuring OWA through 1FA in ActiveSync with IIS Extension

Prerequisites

To use the IIS Extension module:

1. Install Windows Password Provider on the computer with Core Server installed.
2. When using the HTTPS protocol connection, install a client certificate on each Core Server.
3. Install and configure IIS Extension.
4. Configure IIS.

5. Configure the integration of IIS Extension with the business application.
6. If required, specify the optional settings.

Install and configure IIS Extension

To install the module:

1. Run the `Axidian.IIS.Extension-<version number>.x64.en-us.msi` installation file located at `Axidian Access <version number>/Axidian IIS Extension/<version number>` and follow the steps of the installation wizard. If required, you can change the installation folder.

2. Open Windows Registry Editor.

3. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2` section, modify the following parameters:

    - In the `ServerUrlBase` parameter, specify the URL of your Core Server in the format `http(s)://full_dns_server_name/am/core`.
    - In the `IsIgnoreCertErrors` parameter, specify the value *0* or *1*. This parameter stands for verifying the server certificate; with the value *1*, certificate errors are ignored.

4. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID` section, in the `IISHTTPModule` subsection, create the following parameters:

    - The `LSUrl` string parameter. In the parameter value, specify the URL of your Log Server in the format `http(s)://full_dns_server_name/ls/api`.
    - The `ProviderId` string parameter. This is an optional parameter. In the value, specify the identifier of the provider that becomes the default provider at the first login.

Possible `ProviderId` values

- SMS OTP {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}
- Email OTP {093F612B-727E-44E7-9C95-095F07CBB94B}
- Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
- Software OTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
- Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
- Hardware TOTP {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
- Axidian Key (only in the push notification mode) {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}
- Telegram Provider {CA4645CC-5896-485E-A6CA-011FCC20DF1D}

Configure IIS

> **NOTE**
> This instruction describes a configuration example for Exchange 2016.

To configure IIS:

1. In IIS Manager, open the application that will use IIS Extension (for Outlook Web Access, this is *owa*) and go to the **Modules** section.
2. In the **Actions** menu, click **Configure Native Modules**, select the Axidian modules, and click **OK**.

Integrate IIS Extension with a business application

Two-factor authentication must be configured separately for each target application.

To configure the integration:

1. Open Windows Registry Editor.
2. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\IISHTTPModule` section, create a section with the name of the application or site in IIS. The value can be arbitrary.

3. In the created section, create the following parameters:

- `AuthCookie` — a string parameter. The name of the cookie used for authentication in the target application. It is determined experimentally for each application. To get the parameter value from the F12 IE Developer Toolbar console:

    1. In the **Network** section, click **Enable network traffic capturing**.

    2. Authenticate in the application.

    3. Go to the **Details** section, to the **Cookies** tab.

    4. Find the value in the **Key** column.

- `isMFAEnabled` — a DWORD parameter. Enables two-factor authentication.

- `LoginURL` — a string parameter. The relative URL that the application login form data is POSTed to. It must start with a slash (/). The URL is specified relative to the target site.

- `MatchTargetRedirect` — a DWORD parameter. With the value *1*, the page for entering the second factor is displayed before going to the main page. The target page is not saved in the buffer; after the second factor is entered, the transition to the main page occurs (the `TargetURL` parameter).

- `OTPURL` — a string parameter. An alternative URL for sending the second factor authentication form data. By default, the form data is sent to the same URL as the target application form data. The IIS module intercepts it and substitutes the original data if authentication was successful, or does not substitute it if authentication failed and the target application displays its own authentication error. The value must be used if the target application does not treat the form data as erroneous for authentication, or if authentication errors must be shown to the user explicitly. Thus, the value can be left empty.

- `PasswordField` — a string parameter. The value of the `name` attribute of the password field of the application login form.

- `LoginReferer` — a string parameter. Allows you to separate requests from applications that use the same login page. Create a parameter with a unique value for each application.

> ⚠ **IMPORTANT**
> If the server address matches the application name, the request source for the login page contains this application name even if you set different `LoginReferer` values.
>
> **Example**
>
> For example, the server address is *ecp.axidian.local*. You have configured two LoginReferer values — *%2fowa* and *%2fecp*. The login page for both the OWA application and the ECP application will contain the server name part *%2fecp.axidian.local*.
>
> To solve this problem, specify the following in LoginReferer:
>
> - for OWA: *%2fecp.axidian.local%2fowa*
> - for ECP: *%2fecp.axidian.local%2fecp*

- `RedirectToTarget` — a DWORD parameter. The transition to the target page.
- `TargetURL` — a string parameter. The URL of the target page the user gets to after authentication in the application.

> ⚠ **IMPORTANT**
> For Exchange 2013 and 2016, specify `/owa` (without the trailing slash /); for Exchange 2010, specify `/owa/` (with the trailing slash /).

- `UsernameField` — a string parameter. The value of the `name` attribute of the username field of the application login form.

The values of all the parameters — `LoginURL, PasswordField, UsernameField` — are contained in the authentication form of the target application and can be obtained, for example, with the Internet Explorer F12 Developer Tools.

> ⓘ **NOTE**
> If you use the OWA application, disable Basic authentication in Registry Editor. Create a DWORD parameter `IsBasicDisabled` with the value *1* at the following path: `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\IISHTTPModule\IISConfig\owa`.

Optional settings

- Disabling the second factor request
- Changing the localization
- Configuring the session storage in the cache
- Options for writing the domain name for OWA

Example of the module operation

1. Open the OWA application and enter the domain login and password.
2. After the correct input, a window prompting for the second factor appears.
3. After the one-time password is entered successfully, the application opens.

Disable the second factor request

You can cancel the second factor authentication request for users who log in from a trusted network.

You can specify this setting globally for all applications within one IIS Extension installation, or only for a specific application. The global setting is required when applications have overlapping trusted networks.

If you want to override the existing global setting for a specific application or site, duplicate the setting with other networks in the key with the name of the site or application.

To disable the second factor request:

- If you need the global setting, in the registry, in the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\IISHTTPModule` folder, add a *REG_SZ* parameter named `bypassNetworks` and specify the IP addresses with the subnet mask, separated by commas (IPv4 and IPv6 addresses are supported).
- If you want to specify trusted networks for a specific application or site, or override the global setting, in the `iisconfig` key, create a new key with the exact name of the IIS site or application and specify the IP addresses with the subnet mask, separated by commas (IPv4 and IPv6 addresses are supported).

> **NOTE**
> You can specify no more than 252 addresses.

Change the localization

You can configure the localization on the computer with the IIS Extension module installed. To do this:

1. Open Windows Registry Editor.
2. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\IISHTTPModule` section, create the `LanguageId` key of the DWORD type.
3. Set one of the following values in the decimal format:
    - *1033* — English
    - *1049* — Russian

By default, the localization of the installed distribution is used.

Configure the session storage period

For short-term connection losses to Core Server, you can configure the period during which the session can be reused.

During the specified period, the session is stored in the cache and can be reused when a new server connection is requested. The server availability check is not repeated; the cached check result is used instead.

After the storage period expires, the session is terminated. A new session is created at the next server connection request.

To configure the session storage period in the cache:

In Group Policy Editor

1. Open Group Policy Editor.
2. Open **Computer Configuration** → **Administrative Templates** → **Axidian ID** → **Client Connection**.
3. Enable the **Server connection settings** policy.
4. In the policy properties, set the required value for the **Session object hold period (ms)** parameter. The default value is 180,000 ms (3 minutes).

In registry

1. Open Registry Editor.
2. Open **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2**.
3. For the `SessionHoldPeriodMs` parameter of the *DWORD* type, set the required value. The default value is 180,000 ms (3 minutes).

Domain name format for OWA

IIS Extension does not support configuring the login to OWA (Outlook Web Access) by *User name only*.

For IIS Extension to work correctly, write the domain name in the following formats:

- SAM: the domain in the NetBIOS form — `axidian\user`
- UPN: the domain in the DNS form — *user@axidian.local*

If you enable name normalization in Windows Registry Editor (`HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\IISHTTPModule`, the `NameTranslationDisabled=0` parameter of the DWORD type), authorization is impossible if the domain name is entered in a non-standard way:

- SAM: the domain in the DNS form instead of NetBIOS — `axidian.local/user`
- UPN: the domain in the NetBIOS form instead of DNS — *user@axidian*

RDP Windows Logon

The RDP Windows Logon module allows you to implement two-factor authentication with Axidian Access when connecting over the RDP (Remote Desktop Protocol) protocol or in the Remote App application.

The second factor can be one of the following:

- Email OTP
- Hardware OTP
- Hardware TOTP
- Passcode
- Secured TOTP
- SMS OTP
- Software OTP
- the Axidian Key mobile application (in the mode of sending one-time passwords and push notifications with login confirmation)

Prerequisites

To use the RDP Windows Logon module:

1. Enable NLA (Network Level Authentication) for the user.
2. When using the HTTPS protocol connection, install a client certificate on each Core Server.
3. Install and configure RDP Windows Logon.
4. Configure the authentication provider selection on the user side.
5. If required, specify the optional settings.

Install and configure RDP Windows Logon

1. Run the installation file located at `Axidian Access <version number>/Axidian RDP Windows Logon/<version number>` and follow the steps of the installation wizard.
2. Open Windows Registry Editor.
3. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID` section, create the `RemoteAuth` section.

4. In the `RemoteAuth` section, create the `ProviderId` string parameter and set the value corresponding to the provider used.

   Possible `ProviderId` values

   - SMS OTP {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}
   - Email OTP {093F612B-727E-44E7-9C95-095F07CBB94B}
   - Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
   - Software OTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
   - Secured TOTP {F15FD7EC-19EA-4384-846E-A2D0BE149FA2}
   - Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
   - Hardware TOTP {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
   - Axidian Key (only in the push notification mode with login confirmation) {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}

5. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2` section, modify the following parameters:

   - In the `ServerUrlBase` parameter, specify the URL of your Core Server in the format `http(s)://full_dns_server_name/am/core`.

     > ⚠ **IMPORTANT**
     > In the application settings, the URL must not contain a slash (/) at the end of the address.

   - In the `IsIgnoreCertErrors` parameter, specify the value *0* or *1*. This parameter stands for verifying the Core Server certificate; with the value *1*, certificate errors are ignored.

Configure the authentication provider selection for the user

1. In the Windows registry, in the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\RemoteAuth` section, create a DWORD parameter named `IsAuthSelectionEnabled`.

2. Set the value of the `IsAuthSelectionEnabled` parameter to *1*.

If the parameter is not specified or its value is *0*, the authentication provider selection is not offered. In this case, all available authentication methods are displayed.

If `IsAuthSelectionEnabled=1` and a provider is specified in `ProviderId`, the specified provider is selected when the user connects. The user can also select any other supported provider.

Optional settings

- Authentication of users without a license.
- Configuring the session storage period.
- Concurrent operation with Windows Logon.

Example of the module operation

1. Connect to the machine with RDP Windows Logon installed.
2. Specify the user and the domain password and click **OK**.
3. Enter the one-time password.

> **NOTE**
> If the user has no available authentication methods, the message *No available authentication methods. Access denied.* appears and the Remote Desktop session is terminated.

Authentication of users without a license

By default, RDP Windows Logon works with users who have the *AM RDP Windows Logon* license.

Nevertheless, you can configure the authentication of users without a license. To do this:

1. Open Windows Registry Editor.
2. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\RemoteAuth` section, create a `DWORD` parameter named `AllowNonEAUsers`:
   - If the `AllowNonEAUsers` parameter value is *1*, users without the RDP Windows Logon license can authenticate with the domain password (without using Axidian Access technology).
   - If the `AllowNonEAUsers` parameter value is *0* or not specified, authentication is performed only for users with the RDP Windows Logon license. A user without a license cannot authenticate.

Parameter operation scenarios

- If the user is outside the Axidian Access user catalog — the *AllowNonEAUsers* setting applies to the user; the user cannot authenticate if the parameter value is not *1*.
- If the user is in the user catalog without a license, but there is no application in the policy for the user or the user is not included in a policy — the *AllowNonEAUsers* setting applies to the user; such users cannot authenticate if the parameter value is not *1*.
- If the user is in the user catalog without a license and there is an application in the user policy — the *AllowNonEAUsers* setting does not apply to the user. A login attempt with automatic license capture is always performed.

Configure the session storage period

For short-term connection losses to Core Server, you can configure the period during which the session can be reused.

During the specified period, the session is stored in the cache and can be reused when a new server connection is requested. The server availability check is not repeated; the cached check result is used instead.

After the storage period expires, the session is terminated. A new session is created at the next server connection request.

To configure the session storage period in the cache:

In registry

1. Open Registry Editor.
2. Open **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2**.
3. For the `SessionHoldPeriodMs` parameter of the *DWORD* type, set the required value. The default value is 180,000 ms (3 minutes).

In Group Policy Editor

1. Open Group Policy Editor.
2. Open **Computer Configuration → Administrative Templates → Axidian ID → Client Connection**.
3. Enable the **Server connection settings** policy.
4. In the policy properties, set the required value for the **Session object hold period (ms)** parameter. The default value is 180,000 ms (3 minutes).

Concurrent operation with Windows Logon

If the Windows Logon and RDP Windows Logon modules are installed on the same device, configure the policy for Windows Logon.

In registry

1. Go to the device with the RDP Windows Logon and Windows Logon modules installed.
2. Open Registry Editor.
3. Open **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Logon for Windows**.
4. Create a parameter of the *DWORD* type named `CredProvFilter` and set it to the value *2*.

In Group Policy Editor

1. Go to the device with the RDP Windows Logon and Windows Logon modules installed.
2. Open Group Policy Editor.
3. Open **Computer Configuration → Administrative Templates → Axidian ID → Windows Logon**.
4. Enable the **Credential Provider settings** policy.
5. For the **Display of login methods** parameter, select the **All except the password** value.

NPS RADIUS Extension

NPS RADIUS Extension (RADIUS Extension) is an extension module for Microsoft Network Policy Server (NPS, included in Windows Server) that allows you to implement two-factor authentication for RADIUS-compatible services and applications.

In RADIUS Extension, you can authenticate with the following authenticators:

- Email OTP
- Hardware OTP
- Hardware TOTP
- MFA
- Passcode
- Secured TOTP
- SMS OTP
- Software OTP
- Storage SMS OTP
- Telegram Provider (in the mode of sending one-time passwords and push notifications with login confirmation)
- the Axidian Key mobile application (in the mode of sending one-time passwords and push notifications with login confirmation)

> ⓘ **INFORMATION**
> With Axidian Key, two-factor authentication is implemented: the first factor is Windows Password, the second one is a push notification with login confirmation in the Axidian Key mobile application.
>
> Depending on the Axidian Access version, the authenticator can be called either *Axidian Key* or *2FA: Windows Password + Axidian Key (Push)*.

To configure the RADIUS Extension module:

1. Install and configure Network Policy Server (NPS).
2. Install and configure RADIUS Extension.
3. Register the module license in Management Console.

4. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian Radius Providers/<version number>`.
5. Install and configure the authentication providers.
6. Integrate the business application with NPS RADIUS Extension.
7. Add and configure the integrated application in Management Console.
8. If required, specify the additional settings.

Install and configure NPS

NPS RADIUS Extension (RADIUS Extension) is an extension module for Microsoft Network Policy Server (NPS, included in Windows Server). Therefore, before you start installing RADIUS Extension, you must install and configure NPS. Otherwise, the extension does not work.

Install NPS

1. Open Server Manager.
2. In the upper right corner, click **Manage** → **Add Roles and Features**.
3. In the Add Roles and Features Wizard, on the **Installation Type** tab, select **Role-based or feature-based installation**.
4. On the **Server Selection** tab, select the server for the installation.
5. On the **Server Roles** tab, select the **Network Policy and Access Services** role and agree to install the additional components.
6. Go through the wizard steps up to the **Confirm installation selections** tab, leaving the default settings.
7. Click **Install**.

Configure NPS

Add a RADIUS client

1. Start **Network Policy Server**.
2. Expand the **RADIUS Clients and Servers** section.
3. Right-click **RADIUS Clients** and select **New**.
4. In the **New RADIUS Client** window, specify the client settings:
    - In the **Friendly name** field, enter an arbitrary friendly name.

- In the **Address (IP or DNS)** field, enter the address of the client component server.
- In the **Shared Secret** section, specify the secret key for the connection.

> ⓘ **NOTE**
> Specify the shared secret that was set when configuring the client component. If the connection on the client side has not been configured yet, specify an arbitrary secret and remember it. This key is required when configuring the client component connection.

5. Click **OK**.

Add a network policy

1. Start **Network Policy Server**.
2. Expand the **Policies** section.
3. Right-click **Network Policies** and select **New**.
4. In the **Policy name** field, specify an arbitrary name and click **Next**.
5. In the **Specify Conditions** window, add the conditions that are checked when clients connect. To do this, click **Add...** and select a suitable condition. After adding the condition, click **Next**.

> ⓘ **NOTE**
> The *User Groups* condition is added as an example. When adding a group, you need to specify the name of the user group from Active Directory.

6. In the **Specify Access Permission** window, select **Access granted** and click **Next**.
7. In the **Configure Authentication Methods** window, select the authentication methods configured on the client and click **Next**.

> ⚠ **IMPORTANT**
> The authentication methods on the RADIUS server side and on the client side must match, otherwise an authentication error occurs.

> **NOTE**
> If you use MS-CHAP (Challenge Handshake Authentication Protocol) authentication, enable the **Store password using reversible encryption** option in the user account parameters in Active Directory and update the user password.

8. In the **Configure Constraints** and **Configure Settings** windows, leave the default values and click **Next**.

9. In the **Completing New Network Policy** window, check the data and click **Finish**.

Install and configure RADIUS Extension

After installing and configuring NPS, install and configure RADIUS Extension. To do this:

1. Run `AxidianID.EA.RADIUS.Extension-<version number>.x64.en-us.msi` located at `Axidian Access <version number>\Axidian NPS RADIUS Extension\<version number>`.
2. Configure RADIUS Extension in the registry or with group policies.

In registry

1. Open Registry Editor.
2. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2` section, modify the following parameters:

- In the `ServerUrlBase` string parameter, specify the Core Server address in the format `https://full_dns_server_name/am/core/`.

  > **NOTE**
  > For correct operation, we recommend that you install a client certificate on each Core Server.

- In the `IsIgnoreCertErrors` numeric parameter, specify the value *0* or *1*.

  This parameter stands for ignoring the Core Server authentication certificate errors. With the value *1*, certificate errors are ignored.

- If required, you can set the session hold period in the `SessionHoldPeriodMs` numeric parameter. This is the period during which the session can be reused in case of short-term connection losses to Core Server. During the specified period, the session is stored in the cache and can be reused when a new server connection is requested. The server availability check is not repeated; the cached check result is used instead. After the period expires, the session is terminated. A new session is created at the next server connection request.

  The default parameter value is 180,000 ms (3 minutes).

In Group Policy Editor

1. Add the group policy templates.
2. Open Group Policy Editor.
3. Go to **Computer Configuration** → **Administrative Templates** → **Axidian ID** → **Client Connection**.
4. Open the properties of the **Server connection settings** policy.
5. Enable the policy.
6. In the **Core Server URL** field, enter the Core Server address in the format `https://full_dns_server_name/am/core/`.
7. If required, enable the **Ignore certificate errors** option.
8. For the **Session object hold period** setting, leave the default value — 180,000 ms (3 minutes) — or specify a new one.
9. Click **OK**.

For more information about configuring RADIUS Extension with policies and in the registry, see the RADIUS Extension section.

Configure access to applications that have not been added

If applications work through NPS (Network Policy Server) with RADIUS Extension installed, but authentication for them is not performed through Axidian Access, configure access for them in Management Console. Otherwise, login into these applications is not available.

To configure access for RADIUS applications that have not been added to Axidian Access:

1. In the Management Console sidebar, open the **Configuration** section.
2. On the **Integration modules** tab, select **NPS RADIUS Extension**.
3. Allow or deny access in the **For RADIUS applications that are not registered in Axidian Access** setting.

Add and configure an integrated application

In Management Console, you can:

- Add an application integrated through RADIUS Extension.
- Select the login method for integrated applications.
- Configure the attributes for searching repeated authentication requests and perform other optional settings.

Optional settings

Configure the transmission of RADIUS attributes

This setting allows you to add the attributes specified in the NPS server network policy to the `Access-Accept` response.

This is an optional setting; perform it only if RADIUS attributes need to be transmitted to the client.

To configure the transmission of RADIUS attributes:

1. Open **Network Policy Server**.
2. Expand the **Policies** → **Connection Request Policies** section.
3. Select an existing policy or create a new one and open its properties.
4. On the **Settings** tab, go to the **RADIUS Attributes** → **Standard** section.
5. Click **Add**.
6. In the **Add Standard RADIUS Attribute** window, select **Filter-Id** in the **Attributes** list and click **Add**.
7. In the **Attribute Information** window, click **Add**.
8. Make sure that the **Attribute information** parameter is a string and enter a string in the format *IID_CR_AccessAccept_Attributes:<identifier of the required attribute 1>, <identifier of the required attribute 2>*

   > **TIP**
   > If there are several attributes, specify the identifiers separated by commas.

   Example

   ```
   IID_CR_AccessAccept_Attributes:25, 13
   ```

9. Close all the windows, click **Apply**, and restart the NPS service.

Configure access parameters in the NPS policies

In the policy on the NPS server, you can set the following parameters:

- The common login method.
- Login methods for user groups.

This is an optional setting.

To configure the access parameters:

1. Start **Network Policy Server**.
2. Go to the **Policies** → **Connection Request Policies** section.
3. Create a new policy or use an existing one and open its properties.
4. On the **Settings** tab, go to the **RADIUS Attributes** → **Standard** section.
5. Click **Add**.
6. In the **Add Standard RADIUS Attribute** window, select **Filter-Id** in the **Attributes** list and click **Add**.
7. In the **Attribute Information** window, click **Add**.
8. Make sure that the **Attribute information** parameter is a string and enter the corresponding values.

Common login method
If you configure the common login method, the users who are not affected by the **Configuring login methods for user groups** parameter use the specified authentication provider in RADIUS applications. To configure the common login method, enter a value in the format **IID_ModeId_{ProviderId}**, where `ProviderId` is the provider identifier. The `ProviderId` value is unique for each provider.

Identifier values of the providers that support RADIUS authentication

- {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0} — Software TOTP,
- {B772829C-4076-482B-B9BD-53B55EA1A302} — Software TOTP (Challenge\Response),
- {F696F05D-5466-42b4-BF52-21BEE1CB9529} — Passcode,
- {EBB6F3FA-A400-45F4-853A-D517D89AC2A3} — SMS OTP,
- {CA4645CC-5896-485E-A6CA-011FCC20DF1D} — Telegram,
- {093F612B-727E-44E7-9C95-095F07CBB94B} — Email OTP,
- {AD3FBA95-AE99-4773-93A3-6530A29C7556} — Hardware OTP,
- {631F1011-2DEE-47C5-95D8-75B9CAED7DC7} — Hardware OTP (Challenge\Response),
- {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05} — Hardware TOTP,
- {D338344E-7AB7-4D82-9B53-C3678662B153} — Hardware TOTP (Challenge\Response),
- {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68} — Axidian Key,
- {CB3D3B0A-29C6-4BA4-939D-09B126C10C2E} — Passcode + Software TOTP (Challenge\Responce),
- {E5D3185C-9A13-4538-BE8F-D4E1C50A329E} — Passcode + Axidian Key,
- {F15FD7EC-19EA-4384-846E-A2D0BE149FA2} — Secured TOTP,
- {7F3DE86F-59D1-4476-AA5D-F277E5DD5938} — Secured TOTP (Challenge\Response),
- {882C1787-FD32-44A2-BA89-F1F529FBE7AB} — Passcode + Secured TOTP (Challenge\Response),
- {4E32199B-9A21-4CD7-8646-C70C48B55ED9} — Passcode + SMS OTP (Challenge\Response).

> **NOTE**
> Challenge\Response is an authentication method where one side sends a request (the "challenge") and the other side must provide the correct "response" for successful authentication.

Example of an attribute value with Software OTP Provider

```
IID_ModeId_{B772829C-4076-482B-B9BD-53B55EA1A302}
```

For user groups

If you configure the login method for an Active Directory user group, the users of this group use the specified provider for authentication in RADIUS applications.

To configure login methods for user groups, enter a value in the format **IID_Group_ModeId_{DN}_{ProviderId}**, where:

- `DN` is the unique name of the group in Active Directory (Distinguished Name). If you need to specify several groups, separate the values with commas.
- `ProviderId` is the provider identifier.

  The `ProviderId` value is unique for each provider.

Identifier values of the providers that support RADIUS authentication

- {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0} — Software TOTP,
- {B772829C-4076-482B-B9BD-53B55EA1A302} — Software TOTP (Challenge\Response),
- {F696F05D-5466-42b4-BF52-21BEE1CB9529} — Passcode,
- {EBB6F3FA-A400-45F4-853A-D517D89AC2A3} — SMS OTP,
- {CA4645CC-5896-485E-A6CA-011FCC20DF1D} — Telegram,
- {093F612B-727E-44E7-9C95-095F07CBB94B} — Email OTP,
- {AD3FBA95-AE99-4773-93A3-6530A29C7556} — Hardware OTP,
- {631F1011-2DEE-47C5-95D8-75B9CAED7DC7} — Hardware OTP (Challenge\Response),
- {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05} — Hardware TOTP,
- {D338344E-7AB7-4D82-9B53-C3678662B153} — Hardware TOTP (Challenge\Response),
- {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68} — Axidian Key,
- {CB3D3B0A-29C6-4BA4-939D-09B126C10C2E} — Passcode + Software TOTP (Challenge\Responce),
- {E5D3185C-9A13-4538-BE8F-D4E1C50A329E} — Passcode + Axidian Key,
- {F15FD7EC-19EA-4384-846E-A2D0BE149FA2} — Secured TOTP,
- {7F3DE86F-59D1-4476-AA5D-F277E5DD5938} — Secured TOTP (Challenge\Response),
- {882C1787-FD32-44A2-BA89-F1F529FBE7AB} — Passcode + Secured TOTP (Challenge\Response),
- {4E32199B-9A21-4CD7-8646-C70C48B55ED9} — Passcode + SMS OTP (Challenge\Response).

> **NOTE**
> Challenge\Response is an authentication method where one side sends a request (the "challenge") and the other side must provide the correct "response" for successful authentication.

> **NOTE**
>
> If Cyrillic characters are used in the Active Directory group names, set Russian as the language for non-Unicode programs on each Network Policy Server. Without this setting, the members of such groups are denied authentication by the Network Policy Server.

Example of an attribute value with Axidian Key Provider

```
IID_Group_ModeId_{CN=AdminsAxidian,OU=UsersAxidian,OU=Axidian,DC=axidian,DC=local}_{DEE
F0CB8-AD2F-4B89-964A-B6C7ECA80C68}
```

9. Click **Apply** to save the changes made to the policy.
10. Restart the NPS service.

Mobile Device Provisioning

When a user sets up a mail account on a new device via Exchange ActiveSync, the device is automatically placed in quarantine. To start receiving mail, you need to release the device from quarantine. You can do this using the Axidian Mobile Device Provisioning (MDP) module.

System requirements

To configure the Mobile Device Provisioning module:

1. Create and configure a service account for the interaction between MDP and Active Directory.
2. Install the module using the MSI package.
3. Register the module license in Management Console.
4. Perform the settings in Internet Information Services (IIS).
5. Edit the module configuration file.
6. Configure authentication via Identity Provider.
7. Configure the display of the tab in User Console.
8. If necessary, enable logging.

Create and configure a service account

The Mobile Device Provisioning module will access Active Directory through a service account in order to build the list of devices in quarantine. The search by the user's email address is performed on behalf of this account.

To configure the service account:

1. In Active Directory, create a service account.
2. Open the Exchange Control Panel (ECP).
3. In the **Permissions** section, on the **Admin Roles** tab, click + to create a new role group.
4. In the window that opens, specify the following settings:

- In the **Name** field, enter the name of the group.
- In the **Roles** section, add:
    - *Mail Recipients* — the role for receiving device data.

- *Mailbox Search* — the role for receiving the email address. Without this role, the service account will only be able to receive its own email address.
- *Organization Client Access* — the role for granting access to a device and for blocking.

- In the **Members** section, add the service account created earlier.

5. Click **Save**.

Install the module

Run *Axidian.MobileDevProv-v8.2.2.x64.en-us.msi*, located at the path *Axidian Access <version number>/Axidian Mobile Device Provisioning/<version number>*.

When the installation is complete, you will be prompted to generate a new IDP certificate. The certificate is required for authentication via Identity Provider.

Settings in Internet Information Services

After installation, perform the following settings in Internet Information Services:

- Add the service account to the application pool identity.
- Change the version of the CLR runtime.

Add a service account

To add a service account to the application pool, do the following:

1. Open **IIS Manager**.
2. In the left **Connections** menu, expand the node corresponding to your Axidian Access server.
3. Select **Application Pools**.
4. In the list, select *IndeedAM.MDP*. In the menu that opens on the right, select **Edit Application Pool→Advanced Settings**.
5. In the **Advanced Settings** window that opens, in the **Process Model** section, select **Identity** and click  .
6. In the **Application Pool Identity** window that opens, select **Custom account** and enter the user name and password for the created Active Directory service account.
7. Click **OK**. The **Application Pool Identity** window will close.

8. In the **Advanced Settings** window, in the **Process Model** section, select **Load User Profile** and select the value *True* in the drop-down list.
9. Save the changes.

Change the CLR runtime version

To change the CLR runtime version:

1. Open **IIS Manager**.
2. In the left **Connections** menu, expand the node corresponding to your Axidian Access server.
3. Select **Application Pools**.
4. In the list, select *IndeedAM.MDP*. In the menu that opens on the right, select **Edit Application Pool→Basic Settings**.
5. In the **Edit Application Pool** window that opens, in the **.NET CLR runtime version** drop-down list, select **No Managed Code** and click **OK**.

Configure the module configuration

In the MDP configuration file *C:\inetpub\wwwroot\am\mdp\app-settings.json*, make the changes:

1. In the `Exchange` section, in the `ServerUrl` parameter, specify the connection address to the remote Exchange PowerShell server in the format *https://full_dns_server_name/Powershell*.

   Example

   ```
   "Exchange": {
     "ServerUrl": "https://exchange.axidian-test.com/Powershell"
   ```

   Verification in the Exchange Control Panel (ECP)

   In the Exchange Control Panel, open the **Servers** section and go to the **Virtual Directories** tab. In the properties of **PowerShell (Default Web Site)**, make sure that the address specified in the configuration file matches the internal URL.

2. In the `Server` section, in the `Url` parameter, specify the Log Server address in the format *https://full_dns_server_name/ls/api/*.

Example

```
"Server": {
    "Url": "https://amcore.axidian-test.com/ls/api/"
```

3. In the `SAML` section, configure the connection to Identity Provider:
   - In the `LocalServiceProviderConfiguration` section, in the `Thumbprint` parameter, specify the thumbprint of the MDP certificate.

     To get the certificate thumbprint using a PowerShell request, use the command:

     ```
     Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=mdpsp"}
     ```

   - In the `PartnerIdentityProviderConfigurations` section:
     - In the `SingleSignOnServiceUrl` parameter, specify the Single Sign-On Service address in the format *https://full_dns_server_name/am/idp/Account/SsoService*.
     - In `SingleLogoutServiceUrl`, specify the Single Logout Service address in the format *https:// full_dns_server_name/am/idp/Account/SloService*.
     - In `Thumbprint`, specify the thumbprint of the Identity Provider certificate.

     To get the certificate thumbprint using a PowerShell request, use the command:

     ```
     Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=idp"}
     ```

Example

```
//highlight-grey-next-line
"SAML": {
  "$schema": "https://www.componentspace.com/schemas/saml-config-schema-v1.0.json",
  "Configurations": [
    {
      "LocalServiceProviderConfiguration": {
        "AssertionConsumerServiceUrl": "Account/AssertionConsumerService",
        "LocalCertificates": [
          {
            "Thumbprint": "3FD40F59795C9F5977E33E18FCE2D5BB27FDC65C"
          }
        ]
      },
      "PartnerIdentityProviderConfigurations": [
        {
          "Name": "urn:axidianid:saml_idp",
          //highlight-grey-start
          "SingleSignOnServiceUrl": "https://amcore.axidian-
test.com/am/idp/Account/SsoService",
          "SingleLogoutServiceUrl": "https://amcore.axidian-
test.com/am/idp/Account/SLOService",
          //highlight-grey-end
          "PartnerCertificates": [
            {
              //highlight-grey-next-line
              "Thumbprint": "D520DC2E6BEDB1400526842CA88485FFB2F8D757"
            }
```

Configure authentication via Identity Provider

For Axidian Mobile Device Provisioning, two-factor authentication via Identity Provider is provided. If the user has authenticated in User Console, they are automatically authenticated in the module.

How to install and configure Identity Provider

To configure authentication in MDP:

1. Open the Identity Provider configuration file *C:\inetpub\wwwroot\am\idp\app-settings.json*.
2. Make the following changes:
   - In the `CustomAttributes` section, in the `ServiceProvider` parameter, specify the urn of the MDP service and the user attributes from AD, for example:

```
"CustomAttributes": [
  {
    //highlight-grey-next-line
    "ServiceProvider": "urn:axidianid:mobiledeviceprovisioning",
    "Attributes": [
      {
        //highlight-grey-start
        "Name": "user_name",
        "UserNameFormat": "PrincipalName"
        //highlight-grey-end
      }
```

- In the `SAML` section, add the block to the `PartnerServiceProviderConfigurations` section:

```
    {
       "Name": "urn:axidianid:mobiledeviceprovisioning",
       //highlight-grey-next-line
       "SingleLogoutServiceUrl": "https://full_dns_server_name/am/mdp/Account/SLOService",
       "PartnerCertificates": [
         {
           //highlight-grey-next-line
           "Thumbprint": "3FD40F59795C9F5977E33E18FCE2D5BB27FDC65C"
         }
       ]
    },
```

where:

- In the `SingleSignOnServiceUrl` parameter, specify the MDP module address in the format *https://full_dns_server_name/am/mdp/Account/SLOService*.
- In the `PartnerCertificates` parameter, specify the thumbprint of the MDP certificate generated during the module installation.

  To get the certificate thumbprint using a PowerShell request, use the command:

  ```
  Get-Childitem Cert:\LocalMachine\My\ | Where-Object {$_.Subject -eq "CN=mdpsp"}
  ```

Example

```
//highlight-grey-next-line
"SAML": {
  "$schema": "https://www.componentspace.com/schemas/saml-config-schema-v1.0.json",
  "Configurations": [
    {
      "LocalIdentityProviderConfiguration": {
        "Name": "urn:axidianid:saml_idp",
        "SingleSignOnServiceUrl": "https://amcore.axidian-
test.com/am/idp/Account/SsoService",
        "LocalCertificates": [
          {
            "Thumbprint": "D520DC2E6BEDB1400526842CA88485FFB2F8D757"
          }
        ]
      },
      "PartnerServiceProviderConfigurations": [
        {
          "Name": "urn:axidianid:selfservice",
          "SingleLogoutServiceUrl": "https://amcore.axidian-
test.com/am/uc/Account/SLOService",
          "PartnerCertificates": [
            {
              "Thumbprint": "DE67FAABC02FC39BB640B99108918481A3C280EC"
            }
          ]
        },
        {
          "Name": "urn:axidianid:emc",
          "SingleLogoutServiceUrl": "https://amcore.axidian-
test.com/am/mc/Account/SLOService",
          "PartnerCertificates": [
            {
              "Thumbprint": "17BE639E90FA69A5C0F9241E5B0BE6A826604D9C"
            }
          ]
      },
    //highlight-grey-start
        {
          "Name": "urn:axidianid:mobiledeviceprovisioning",
          "SingleLogoutServiceUrl": "https://amcore.axidian-
test.com/am/mdp/Account/SLOService",
          "PartnerCertificates": [
            {
              "Thumbprint": "3FD40F59795C9F5977E33E18FCE2D5BB27FDC65C"
            }
          ]
        },
    //highlight-grey-end
```

Configure the display of the module in User Console

The module is an additional tab in User Console — **Exchange Mobile Devices**, in which the user can release a device from quarantine on their own.

By default, the Axidian Mobile Device Provisioning module is not displayed as a separate tab in User Console.

To configure the display of the module:

1. Open the User Console configuration file *C:\inetpub\wwwroot\am\uc\Web.config*.
2. Make the following changes:
    - Add the following line to the `configSections` section, if it is not there:

      ```
      section name="amMobileDeviceProvisioning"
      type="IndeedID.Web.SelfService.Settings.MobileDeviceProvisioningSettings"
      ```

    - Find and change the line:

      ```
      amMobileDeviceProvisioning isEnabled="false"
      url="MOBILE_DEVICE_PROVISIONING_URL/device/embed"
      ```

        - In the `isEnabled` parameter, specify the value *true* to configure the display of the tab in User Console.
        - In the `url` parameter, specify the address of the server with the installed Mobile Device Provisioning module in the format *https://full_dns_server_name/am/mdp/device/embed*.

Example

```
<configuration>
<configSections>
  <section name="amAuthServer"
type="IndeedID.Web.Shared.Utils.ConfigReader.AuthServerSettings, IndeedID.Web.Shared,
Version=1.0.0.0, Culture=neutral" />
  <section name="amCulture" type="IndeedID.Web.Shared.Utils.ConfigReader.CultureHandler,
IndeedID.Web.Shared, Version=1.0.0.0, Culture=neutral" />
  <section name="amAuthentication"
type="IndeedID.Web.Shared.Utils.ConfigReader.AuthConfigReader" />
  <section name="amProviderImages"
type="IndeedID.Web.Shared.Utils.ConfigReader.ProviderImagesHandler, IndeedID.Web.Shared,
Version=1.0.0.0, Culture=neutral" />
  <section name="amApplicationSettings"
type="IndeedID.Web.Shared.Utils.ConfigReader.ApplicationSettingsHandler" />
//highlight-grey-next-line
    <section name="amMobileDeviceProvisioning"
type="IndeedID.Web.SelfService.Settings.MobileDeviceProvisioningSettings" />
</configSections>
<amAuthServer Url="https://amcore.company.com/am/core/" />
<amCulture isEnabled="true" />
<amProviderImages configSource="Config\providerImages.config" />
<amAuthentication mode="Saml" loginUrl="https://amcore.company.com/am/idp/"
identityProviderCertificateThumbprint="D520DC2E6BEDB1400526842CA88485FFB2F8D757"
serviceProviderCertificateThumbprint="DE67FAABC02FC39BB640B99108918481A3C280EC"
enableLogout="true" />
<amApplicationSettings configSource="Config\applicationSettings.config" />
//highlight-grey-next-line
<amMobileDeviceProvisioning isEnabled="true" url="https://amcore.axidian-
test.com/am/mdp/device/embed" />
<appSettings>
```

⚠ **IMPORTANT!**

Identity Provider is used for authentication in the MDP module. If User Console and Identity Provider are on different machines and have different domain names, add the `Content-Security-Policy` parameter to the `customHeaders` section in the Identity Provider configuration file *Web.config*.

Example

```
<customHeaders>
<add name="Content-Security-Policy" value="frame-ancestors 'self' https://*.domain.local"
/>
</customHeaders>
```

Configure logging

To learn how to configure log collection for the MDP module, read the Collecting logs of server components section.

The following module events are displayed in the Axidian Access event log:

- The user successfully released the mobile device from quarantine.
- Failed to release the mobile device from quarantine
- The user blocked the mobile device
- Failed to block the mobile device

For more details, see the Events section.

Enterprise Single Sign-On

Axidian Access Enterprise Single Sign-On (ESSO, Enterprise SSO) implements the Single Sign-On technology for accessing the information systems of an organization. The module stores user passwords centrally and substitutes the password automatically in a hidden form when a user logs in to an application or performs other actions that require authentication.

Fields are filled in automatically after the user identity is confirmed with an authenticator. This way, Enterprise Single Sign-On relieves users of the need to memorize, write down, store, and manually enter passwords to log in to an application.

Since SSO profiles are stored centrally, users can access applications from any computer where the component is installed.

Authentication technologies

In Enterprise SSO, you can authenticate with the following authenticators:

- Email OTP
- Futronic Provider
- Hardware OTP
- Hardware TOTP
- IronLogic Z2 USB Provider
- MFA Provider
- OMNIKEY Provider
- Passcode
- Secured TOTP
- Smart Card Provider
- SMS OTP
- Software OTP
- Storage SMS OTP
- the Axidian Key mobile application (in the mode of sending one-time passwords and push notifications with login confirmation)

Supported applications

The Enterprise Single Sign-On access technology is used with Windows and web applications and is configured without interfering with either the server or the client part of the target application. To support a new application, you need to create a special template in the XML format. The template defines the application forms that are controlled. Access control is performed as a repeated authentication request, filling in fields with account data, activating the required controls, and writing an event to the log.

For information about how to add and configure supported applications in Management Console, see ESSO module.

Prerequisites

To use Enterprise SSO:

1. Install the ESSO Agent module on the client computer.
2. Configure the connection to Core Server.
3. To work with web applications, configure the ESSO browser extension.
4. Create a template of the target application using the ESSO Template Wizard utility.
5. Add and configure the application in Management Console.

Install ESSO Agent

> **NOTE**
> The module must be installed on user computers. Local administrator rights are required for this.

To install ESSO Agent:

1. Run the installation file `Axidian Access <version number>/Axidian ESSO Agent/<version number>/AxidianID.Enterprise SSO.Agent.msi` and follow the instructions of the installation wizard.
2. After the module installation is complete, the system must be restarted. Click **Yes** to restart immediately, or **No** to do it manually later.

To install ESSO Agent on user computers automatically, you can use Microsoft Group Policy or any other tool that allows you to distribute and install MSI packages in bulk, such as Microsoft System

Center Configuration Manager.

Configure ESSO Agent

You can configure Enterprise SSO either in Registry Editor or with group policies.

Settings applied with group policies take priority over the settings specified manually in the registry. You can view the group policy settings in the registry at `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\SrvLocator2`.

In registry

1. Open Windows Registry Editor on the user computer.
2. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2`.
3. Open the properties of the `ServerUrlBase` string parameter.
4. In the **Value** field, specify the URL of your Core Server in the format `http(s)://full_dns_server_name/am/core/`.

> **NOTE**
> When using the HTTPS protocol connection, install a client certificate on each Core Server.

In Group Policy Editor

1. Add the `AxidianID.ServerUrl.admx` policy to the computer with ESSO Agent installed. Group policy templates are located in the `Axidian Access <version number>\Misc\ADMX Templates` folder.
2. Open Group Policy Editor.
3. Open **Computer Configuration** → **Administrative Templates** → **Axidian ID** → **ClientConnection**.
4. Enable the **Server connection settings** policy.
5. In the **Core Server URL** field, specify the address of your Core Server in the format `http(s)://full_dns_server_name/am/core/`.

Configure the ESSO browser extension

When working with web applications, ESSO uses an extension for the Internet Explorer, Mozilla Firefox, and Google Chrome browsers. To log in to a web application, it is enough to open the service page — the extension performs authentication automatically. You do not need to start ESSO and select the application for SSO authentication manually.

Internet Explorer

The Axidian Access SSO Helper extension for Internet Explorer is installed together with ESSO Agent. To make the extension work, enable it in the browser.

If the ESSO Agent extension does not appear in the browser after the installation, go to **Internet options→Advanced→Browsing** and enable the **Enable third-party browser extensions** option.

> ⓘ **NOTE**
> For server operating systems, additionally go to **Server manager→Local server** and disable **Internet Explorer Enhanced Security Configuration** (IE ESC).

To configure the extension settings in Internet Explorer in bulk, use Windows group policies.

The policies are located in *Local Group Policy Editor (gpedit.msc)*, in the **User Configuration→Administrative Templates→Windows Components→Internet Explorer** section.

1. Enable the *Automatically activate newly installed add-ons* policy so that the extension is activated automatically.
2. To prevent users from disabling add-ons, enable the *Do not allow users to enable or disable add-ons* policy.

Mozilla Firefox

The Axidian ID ESSO extension is installed together with the ESSO Agent module. To make the extension work, enable it in the browser.

To install the extension:

1. In the browser, go to the **Extensions and Themes** section.

2. In the upper right corner, click the settings icon and select **Install Add-on From File**.

3. Go to the `C:/Program Files (x86)/Axidian-Id/Enterprise SSO/Mozilla FireFox Web Extension/Content/axidian_id_esso-vX.X.xpi` folder and click **Open**.

4. After the extension is downloaded, click **Add**.

Google Chrome

The Axidian ID ESSO extension is installed either manually or automatically with Microsoft Windows group policies.

When installing manually from the Chrome Web Store, the extension must be downloaded and enabled for each computer user.

The extension distributed with group policies is installed for all computer users. Internet access is not required for the installation.

To install the extension:

From the Chrome Web Store

1. Install ESSO Agent on a computer with internet access.
2. Start Google Chrome.

The extension is downloaded automatically within 1 minute. After the extension is downloaded, internet access is not required.

To make the extension work, enable it in the browser.

> ⓘ **NOTE**
> If the extension was removed from the browser manually, download it from the Chrome Web Store to install it again.

In Group Policy Editor

1. Start IIS Manager.

2. In the **Connections** pane, expand the **server name** → **sites** → **Default Web Site** node.

3. Right-click **Default Web Site** and select **Add Application**.
4. In the window that opens, specify the following settings:
    - In the **Alias** field, specify an arbitrary application name, such as *chromeplugin*.
    - For the **Application pool** setting, select **DefaultAppPool**.
    - In the **Physical path** field, select the directory for the application.
5. Click **OK**.
6. In the application settings, open **Authentication** and enable **Anonymous Authentication**.
7. Place the *lcjenjmcehnkfkghcflkfialplejjkdj.crx* and `update.xml` files from the `Axidian Access <version number>\Axidian ESSO Agent\<version number>\Misc\Chrome.WebExt` distribution into the application directory.
8. In IIS Manager, go to the **Default Web Site** node.
9. In the **MIME Types** settings, add a new type with the .CRX extension and the `application/chrome` description.
10. Configure a secure HTTPS connection for the *chromeplugin* application and make sure that it can be accessed from user computers.
11. Add the ADM/ADMX templates to the local or central storage of administrative templates of the domain controller.
12. Open **Group Policy Management** and create a new group policy object with an arbitrary name.
13. Right-click the created group policy object and select **Edit**.
14. In the window that opens, go to **Computer Configuration→Administrative Templates→Google→Google Chrome→Extensions**.
15. Enable the *Configure the list of force-installed apps and extensions* policy.
16. In the policy properties, in the **Options** section, click **Show**.
17. In the window that opens, specify the extension identifier from the CRX file name and the path to the XML file in the *chromeplugin* directory.
18. Enable the *Configure extension, app, and user script install sources* policy.
19. Specify the address of the server where the *chromeplugin* application is deployed.

20. Add the user computers with ESSO Agent installed to the scope of the group policy object.

    The extension is downloaded and installed automatically after the group policy is applied to the user computer. It may take several minutes for it to appear in the list of the extensions installed in the browser. An extension installed with group policies is marked with the corresponding symbol.

Create application templates

The ESSO Template Wizard utility is designed for preparing XML templates of Win32 and web applications integrated with the ESSO Agent module.

The utility processes the form of the target application and generates a form snapshot. This is a set of data describing the form elements, such as the title name and the identifiers of buttons and input fields. Based on the data contained in the template, ESSO Agent tracks the start of the target application on the user workstation and, depending on the settings for the specific user, grants or denies access.

> **NOTE**
> The resulting snapshot is a draft for creating the application template. After the snapshot is generated, you need to edit it manually.

Install ESSO Template Wizard

Install ESSO Template Wizard on the workstations where the applications you need to prepare a template for are installed.

To do this, run the `Axidian Access <version number>/Axidian Template Wizard/<version number>/AxidianID.ESSO.Template.Wizard.msi` file and follow the instructions of the installation wizard.

After the installation is complete, the ESSO Template Wizard components — Web Template Wizard and Win32 Template Wizard — appear in *All Programs* → *Axidian* → *Enterprise SSO*.

Collect information about the application

Before creating an application template, collect the following information about it:

1. The set of target application windows that the ESSO template will process. You can be guided by the main operations that a template can perform:
    - Login to the application
    - Unlocking
    - Password change in the application

In some cases, the user scenario implies giving up the password — the user does not know their password and uses an authenticator or transparent authentication to work with the target application. In this case, all windows where the user is prompted to enter the application password must be included in the list of processed windows. Besides the target windows, pay attention to the controls (interface controls) that must be processed by the template.

2. The set of ways to start the application, such as through shortcuts, command files, or executable files. This information is required to configure the quick start of the application.

3. The set of executable files that create the processes displaying the application graphical interface — the windows that ESSO Agent must respond to. For each such file, you need to know the name, description, and location. If the application has only one executable file, the information that Template Wizard collects during its operation is sufficient.

Create a template

After you define the list of required windows and operations, you can start creating the template snapshot or snapshots. Several snapshots may be required for one window. Also, additional snapshots may be required while debugging the template. Therefore, access to the workstation with the target application and ESSO Template Wizard installed is required throughout the entire template creation process.

For Win32 applications

1. Start the Enterprise SSO — Win32 Template Wizard application.

2. Start the target application you need to create a template for and go to its login form.

3. Press **Ctrl+Shift+T**.

4. The template generation settings window opens in Template Wizard. Specify the required parameters in it.

Form control depth settings

- **Form control view depth**.

  With this parameter, you can set the nesting depth of the form controls that Template Wizard uses to build the template. This parameter is set manually when creating a template. If the parameter is *0*, Template Wizard collects information only about non-nested controls. If the parameter is *1*, Template Wizard collects data about non-nested controls and about the controls of the first nesting level, and so on.

- **Control depth of the selected form**.

  This parameter displays the maximum nesting depth of controls on the selected form — how many nesting levels of controls exist on the form. When selecting the value of the **Form control view depth** parameter, we recommend that you be guided by this parameter. Template Wizard calculates the control depth value for the selected form individually for each form at the moment the target application form is defined.

Target application form settings

This set of settings allows you to define the combination of application form attributes that ESSO Agent takes into account when searching for the target window among the multiple windows of the target application.

Title	If this setting is enabled, the form title value is taken into account when tracking the target form of the target application. A fixed title value is added to the template for comparison. If the setting is disabled, the form title value is not taken into account when defining the target application form. We recommend that you use this parameter in the template, except when the title changes dynamically.
Window class name	If this setting is enabled, the window class name for this form is taken into account when tracking the target form of the target application. The window class name is set at the application development stage. A fixed window class name value is added to the template for comparison. If the setting is disabled, the window class name is not taken into account when defining the target application form. We recommend that you use this parameter in the template, except when the class name changes dynamically.
Size	If this setting is enabled, strictly fixed window dimensions — width and height in pixels — are taken into account when tracking the target form of the target application. A fixed form size value is added to the template for comparison. If the setting is disabled, the window size is not taken into account when defining the target application form. Use this parameter only when the other parameters do not describe the target application form accurately enough.

Target application form control settings

This set of settings allows you to define the combination of attributes of the controls present in the application window that ESSO Agent takes into account when searching for the target window among the multiple windows of the target application.

CtrId	If this setting is enabled, the identifiers of all controls that are part of the window are taken into account when tracking the target form of the target application. These identifiers are set at the application development stage. The values of all control identifiers are fixed in the template. If the setting is disabled, the control identifiers are not taken into account when defining the target application form. We recommend that you use this parameter, except when the control identifiers in the target form change dynamically.
Window class name	If this setting is enabled, the window class name for this form is taken into account when tracking the target form of the target application. The window class name is set at the application development stage. A fixed window class name value is added to the template for comparison. If the setting is disabled, the window class name is not taken into account when defining the target application form. We recommend that you use this parameter in the template, except when the class name changes dynamically.

Text	Enabling this setting allows you to take into account the text contained in the controls in the application window (input fields, drop-down lists, and so on) when tracking the target form of the target application. The parameter is taken into account for all controls that are part of the window. The values of all text strings for the controls are fixed in the template. If the setting is disabled, the control text is not taken into account when defining the target application form. **IMPORTANT**: When creating a template, pay attention to changes in the control text. Under various circumstances — for example, when the language in the system or in the application changes, the application version changes, or the user is switched — the text in a control may change. This may interfere with the application tracking procedure. Do not use dynamically changing text in the template, or take into account all possible variants of this parameter.
Size	If this setting is enabled, strictly fixed dimensions of all controls in the window — width and height in pixels — are taken into account when tracking the target form of the target application. A fixed size value is added to the template for comparison. If the setting is disabled, the control dimensions are not taken into account when defining the target application form. Use this parameter only when the other parameters do not describe the properties of the target application form controls accurately enough.
Position	If this setting is enabled, a strictly fixed position — the position of the upper left corner of the control relative to the upper left corner of the container holding this control, in pixels — is taken into account for all controls in the window when tracking the target form of the target application. A fixed position value is added to the template for comparison. If the setting is disabled, the control position is not taken into account when defining the target application form. Use this parameter only when the other parameters do not describe the properties of the target application form controls accurately enough.

Visible	If this setting is enabled, the visibility attribute — whether a control present in the form is visible to the user — is taken into account for all controls that are part of the window when tracking the target form of the target application. The visibility property values for the controls are fixed in the template. If the setting is disabled, the control visibility is not taken into account when defining the target application form.
Enable	If this setting is enabled, the availability attribute — whether a control is available for user actions — is taken into account for all controls that are part of the window when tracking the target form of the target application. The availability property values for the controls are fixed in the template. If the setting is disabled, the control availability is not taken into account when defining the target application form.

Filling algorithm settings

In this section, you set the filling algorithm that is used when substituting the username and password into the target ESSO application form.

None	The filling algorithm is not generated in the template. The form is not filled in automatically.
Based on SetText	If this setting is selected, Template Wizard generates the commands for filling in the username input field, filling in the password input field, and clicking the **OK** button. Each command contains the data of the control it must be applied to. The form controls are filled in with the standard SetText method. We recommend that you use this filling algorithm.
Input emulation	If this setting is selected, Template Wizard generates the commands for setting the focus in the username input field, filling in the username input field, filling in the password input field, clicking the **OK** button, switching between controls, and pause commands. The filling itself is performed by emulating keyboard keystrokes. The focus is switched between controls by emulating the Tab keystroke. The filling algorithm also contains 50-millisecond pause commands between the filling and switching commands. Recommended for use in templates for applications where using the SetText method is impossible or difficult.

NOTE

When creating a template for a specific application, set only the set of settings that distinguishes this application from others. Before generating the template, it is important to understand which attributes are used to define the application login form and which filling algorithm must be applied.

5. Click **Generate**.

6. The **Form control definition** window opens. Go to the target application window, to the required control, and press **Ctrl+Shift+T**. Following the Template Wizard prompts, perform these actions for each selected control.

   After the controls are defined, the template draft is created.

7. Enable the **Save in UTF-16** option and click **Save**.

8. In the window that opens, select the save location and specify the file name of the template draft. Click **Save**.

   The draft is saved in the XML format.

9. In the Template Wizard window, click **Exit** and close the target application.

10. Open the draft in **Notepad** or another text editor that supports the XML format.

11. Check the path to the target application in the `paths` section. If required, you can add an additional path where the application is available.

Example

```
    <paths>

<path>C:\Users\Admin\Desktop\TestApps\TestApp\TestAppWin\AxidianID.SSO.Test.App.exe</path>
// highlight-green-next-line
        <path>C:\Users\Admin\Desktop\AxidianID.SSO.Test.App.exe</path>
    </paths>
```

12. In the `userDescription` parameter, specify the name of the target application. This is the name the user sees in the ESSO Agent window on their device.

Example

```
<userDescription>Test.App.Win</userDescription>
```

13. In the `cmdLine` parameter, check the command for the quick start of the application. This must be the path to the application executable file.

Example

```
    <applicationDetails>
// highlight-green-next-line
        <cmdLine>C:\Users\Admin\Desktop\AxidianID.SSO.Test.App.exe</cmdLine>
    </applicationDetails>
```

14. If **Input emulation** was selected as the filling algorithm when generating the template draft, check which field the cursor is initially placed in when the application starts and in which order it moves through the other controls. Based on the results, edit the form filling algorithm in the `fillingAlgorithm` section of the template if required.

15. Save the modified template file with the .APP extension.

For web applications

1. Start the Enterprise SSO — Web Template Wizard.

2. Start the target application in the Internet Explorer browser.

3. On the target page of the application, press **Ctrl+Shift+T**.

The **Form attribute definition** window of the wizard opens.

If the **Ctrl+Shift+T** key combination does not work, go to **Internet options**. In the **Advanced→Browsing** section, check whether the **Enable third-party browser extensions** option is enabled.

4. Go to the target page, select a small area containing the username and password input fields and the buttons, and press **Ctrl+Shift+T**.

At this moment, the wizard defines the HTML DOM tree nodes. When the operation is complete, the number of defined nodes is displayed at the bottom of the wizard window, in the **Page HTML DOM tree nodes defined** line.

> ⓘ **NOTE**
> This step may take some time. The duration of the wizard operation depends on the number of objects in the processed area of the web page. Do not rush to close the page or the wizard window.

5. After the page attributes are defined, click **Generate template** and specify the generation parameters.

Target application form settings

This set of settings allows you to define the combination of application web page attributes that ESSO Agent takes into account when searching for the target page among the multiple pages loaded in the browser.

Page URL	If this setting is enabled, the page URL is taken into account when tracking the target web page of the ESSO application. A fixed URL value is added to the template for comparison. If the setting is disabled, the URL is not taken into account when defining the target web page. We recommend that you use this parameter in the template, except when the URL changes dynamically. For such pages, either do not use this parameter or use a regular expression.
Page title	If this setting is enabled, the page title value is taken into account when tracking the target web page of the ESSO application. A fixed title value is added to the template for comparison. If the setting is disabled, the title value is not taken into account when defining the target web page. We recommend that you use this parameter in the template, except when the title changes dynamically. For such pages, either do not use this parameter or use a regular expression.

Target application form control settings

This set of settings allows you to define the combination of attributes of the web controls present on the page that ESSO Agent takes into account when searching for the target page among the multiple pages loaded in the browser.

Tag name	If this setting is enabled, the tag name is taken into account for each web element that is part of the DOM tree of the page fragment selected at the template generation preparation stage, when tracking the target web page of the ESSO application. A fixed tag name value is added to the template for comparison. If the setting is disabled, the tag name for web elements is not taken into account when defining the target application form. We recommend that you use this parameter in the template, except when the tag name changes dynamically.
Text	If this setting is enabled, the text is taken into account for each web element that is part of the DOM tree of the page fragment selected at the template generation preparation stage, when tracking the target web page of the ESSO application. A fixed text value is added to the template for comparison. If the setting is disabled, the text component value for web elements is not taken into account when defining the target application form. When using this parameter, we recommend that you check the resulting template — the text array in the template may exceed the expected size, which will cause problems with tracking the target web page and complicate template editing.

id attribute	If this setting is enabled, the value of the id attribute is taken into account for each web element that is part of the DOM tree of the page fragment selected at the template generation preparation stage, when tracking the target web page of the ESSO application. Fixed id attribute values are added to the template for comparison. If the setting is disabled, the id attribute value for web elements is not taken into account when defining the target application form. Recommended for use.
name attribute	If this setting is enabled, the value of the name attribute is taken into account for each web element that is part of the DOM tree of the page fragment selected at the template generation preparation stage, when tracking the target web page of the ESSO application. Fixed name attribute values are added to the template for comparison. If the setting is disabled, the name attribute value for web elements is not taken into account when defining the target application form. Recommended for use.
Other attributes	If this setting is enabled, a set of values for the specified list of attributes is taken into account for each web element that is part of the DOM tree of the page fragment selected at the template generation preparation stage, when tracking the target web page of the ESSO application. The attributes to track must be selected from the drop-down list. To open the list, click **...**. Fixed values of the specified attributes are added to the template for comparison. We recommend that you use this parameter when the other parameters cannot describe the form element unambiguously.

Style attributes	If this setting is enabled, a set of values for the specified list of style attributes is taken into account for each web element that is part of the DOM tree of the page fragment selected at the template generation preparation stage, when tracking the target web page of the ESSO application. The style attributes to track must be selected from the drop-down list. To open the list, click **....** Fixed values of the specified style attributes are added to the template for comparison. We recommend that you use this parameter when the other parameters cannot describe the form element unambiguously.
If there is an id, substitute everything else	If a web element has the id attribute, all other attributes for this element are not used in the template. You can apply this setting if the id attribute is the main parameter used to define the target page.
Use text only when there are no others	The text attribute is used if no other specified attributes are found for this element. Otherwise, the text attribute is not used in the template. Recommended for use.

Filling algorithm settings

In this section, you select the filling algorithm that is used when substituting the username and password into the target ESSO application web page.

None	The filling algorithm is not generated in the template. The form is not filled in automatically.
Based on fillWebCtrl	If this setting is selected, Template Wizard prompts the user to specify 3 controls on the web page with the **Ctrl+Shift+T** key combination and generates 3 commands in the filling algorithm for these elements: • the first command fills the web control with the user login value; • the second command fills the web control with the ESSO account password value; • the third command clicks the **OK** button. Each command contains the data of the element on the page that this command works with. This is an analogue of the SetText algorithm for Win32 applications.

> **NOTE**
> When creating a template for a specific application, set only the set of settings that distinguishes this application from others. Before generating the template, it is important to understand which controls are used to define the login form of the web application.

6. Click **Generate**.

If a filling algorithm was set when configuring the template generation, the **Form control definition** window opens.

7. On the target application page, go to the required control and press **Ctrl+Shift+T**. Following the Template Wizard prompts, perform these actions for each selected control.

8. Enable the **Save in UTF-16** option and click **Save**.

9. In the window that opens, select the save location and specify the file name of the template draft. Click **Save**.

10. The draft is saved in the XML format. In the Template Wizard window, click **Exit** and close the target application.

11. Open the draft in **Notepad** or another text editor that supports the XML format.

12. If the following line is present at the beginning of the file, delete it:

```
<?xml version="1.0" encoding="UTF-16"?>
```

13. Check the address of the target application in the `url` parameter.

Example

```
<componentWeb>
  <url>https://amcore.axidian.com/TestAppWeb/logon.html</url>
</componentWeb>
```

The address value can be set as text or as a regular expression. If the address of the target application changes dynamically, use a regular expression. In this case, the `url` parameter must contain the `isRegex` attribute with the value *1*. With the value *0*, the application address is defined by text.

Regular expression example

```
<componentWeb>
  <url isRegex="1">https:\/\/amcore\.axidian\.com\/TestAppWeb\/logon\.html(.*)</url>
</componentWeb>
```

14. In the `userDescription` parameter, specify the name of the target application. This is the name the user sees in the ESSO Agent window on their device.

Example

```
<userDescription>Test.App.Web</userDescription>
```

15. In the `cmdLine` parameter, check the command for the quick start of the application. This must be the browser start command and the address of the application page.

Example

```
    <applicationDetails>
// highlight-green-next-line
        <cmdLine>"%ProgramFiles%\\Internet Explorer\\iexplore.exe"
"https://amcore.axidian.com/TestAppWeb/logon.html"
</cmdLine>
    </applicationDetails>
```

16. Save the modified template file with the .APP extension.

Useful links

- Enterprise SSO templates
- Enterprise SSO templating
- Preparing an SSO template to work in Firefox
- Regular expressions in SSO templates
- Delays when switching between controls during input emulation
- The template engine does not respond to pages in IE

Work with ESSO Agent

To access applications through Enterprise SSO using an authenticator, you must:

- configure the account of the application;
- make sure that the user who needs access is added to the policy with the target application;
- register an authenticator.

Start ESSO Agent

1. Log in to the workstation. A pop-up message about opening a Single Sign-On session appears in the taskbar.
2. ESSO Agent starts automatically when you log in to the operating system. To start it manually, select **Axidian→Enterprise SSO Agent** in the **Start** menu.

Select the target application

1. Open the ESSO Agent application in one of the following ways:
    - Press [**Ctrl**]+[**Alt**]+[**Q**].
    - Double-click the application icon in the Windows notification area.
    - Right-click the icon in the Windows notification area and select **Quick start** in the application context menu.
2. The **Select an application** window displays the applications allowed to be started with ESSO Agent. The window appearance is defined by the ESSO application settings. Select the required application.

    > ⓘ **NOTE**
    > An application is not displayed in the list if ESSO Agent cannot find its executable file at the user workstation.

    If an application contains several components, the executable files of each component are grouped under a single name in the quick launch panel for more convenience.

3. If several SSO accounts are configured for one target application, select the account after you select the application.

Log in to an application with an authenticator

To log in to an application with an authenticator:

1. Select the username and the login method. By default, the operating system offers the last method used. Follow the on-screen instructions. If you have several registered authenticators, you can use any of those.

2. If several authenticators are registered, select any of those. To select an authenticator, click **Switch authentication method**.

3. After authentication, the login window of the target application appears. The *Username* and *Password* fields are automatically filled in with the data specified when the SSO account was created, and the login is performed.

   Besides automatic filling in of credentials, manual entry of the username and password is also supported. The username and password entered at the first login to the application are stored and filled in automatically at the next login.

   If authentication is cancelled, the login is not performed: the current application form or the application itself is closed.

4. If the application start settings allow starting the application only with ESSO Agent and authentication is required to log in, cancelling authentication results in an error message.

Access to target applications is regulated by administrative settings. Some applications can be prohibited from starting by the ESSO administrator. Prohibited applications are not available in the **Select an application** window. An attempt to start a prohibited application results in the error message *A device attached to the system is not functioning*.

Switch the user

When using ESSO Agent, you can select a different user to access an application.

To switch the user:

1. Log in to the operating system.

2. Right-click the ESSO Agent icon in the Windows notification area.
3. In the application context menu, select **Switch user...**
4. In the **Login to SSO** window, enter the name of the account you want to use to log in to the application.
5. Authenticate with one of the authenticators registered for the selected user. If authentication is successful, ESSO Agent opens an SSO session for the selected user.

Update data

If the SSO profile data changes, you need to load the changes into ESSO Agent. To do this:

1. Right-click the ESSO Agent application icon in the Windows notification area.
2. In the context menu that opens, select **Update data**.

The data update status is displayed in pop-up windows.

Examples of pop-up windows

- Updating data:
- New data received:
- Data update error:

Deactivate hotkeys

If required, you can deactivate the use of hotkeys.

The following key combinations are enabled by default:

- [**Ctrl**]+[**Alt**]+[**Q**]: Opens the quick start window.
- [**Ctrl**]+[**Alt**]+[**U**]: Performs an ESSO data update.
- [**Ctrl**]+[**Alt**]+[**R**]: Performs forced processing of the application form (re-matching).

To deactivate hotkeys:

1. Right-click the ESSO Agent application icon in the Windows notification area.

2. In the context menu that opens, select **Deactivate hotkeys**.

Login when the network is unavailable

You can log in to an application with an authenticator even without a physical network connection. Logging in with a cached authenticator is similar to logging in with an authenticator when the network is available.

To configure login when the network is unavailable:

1. Configure session caching with group policies or in the registry.
2. Enable caching and configure the authenticator validity period in Management Console.
3. Configure the session storage period in the cache.

Prerequisites

Login when the network is unavailable is possible in the following cases:

- Your account has been granted permission to cache authenticators.
- Authenticators were cached at the first login to the application with an authenticator.
- A login with a cached authenticator was performed while the network was available.

Configure session caching

When there is no connection to Core Server, users can authenticate in ESSO Agent in the offline mode. For this, a session cached on the user computer is used.

To configure login with a cached session, use the **Session public key settings** policy in Group Policy Editor:

1. Add the `AxidianID.SessionPublicKey.admx` policy file from the `Axidian Access <version number>\Misc\ADMX Templates` folder to the computer with Core Server.
2. Open Group Policy Editor.
3. Open **Computer Configuration** → **Administrative Templates** → **Axidian ID** → **Cache server**.
4. Enable the **Session public key settings** policy.
5. In the policy properties, in the **Session public key** field, enter the value of the public key generated during the Core Server installation. You can find the public key value in the `SessionPublicKey.pub` file or in the Core Server `Web.config` configuration file, in the `sessionEncryptionSettings` section, in the `publicKey` parameter.

6. Go to **Computer Configuration → Administrative Templates → Axidian ID → Client Connection**.
7. Enable the **Axidian EA\ESSO server availability check** policy.
8. In the policy parameters, set the values for the **Check result waiting timeout (ms)** and **Check result lifetime (ms)** settings.

Enable caching and configure the authenticator validity period

You can enable caching and set the validity period of cached authenticators in the application settings of a policy or a user in Management Console. If the validity periods are not specified, caching is valid indefinitely.

If the **Number of days** parameter is set for cached authenticators, their validity period is counted from the time of the last login through Enterprise SSO with the network available, using any authenticator except Windows Password.

For example, if the period is set to 10 days and the login to the application was performed at 10:00 on 12.04.2025, the cached authenticators are valid from 10:00 on 12.04.2025 until 10:00 on 22.04.2025.

If the validity period of cached authenticators has expired and Core Server is unavailable, the following message appears when you try to log in to the application: *Login error. The user data caching period has expired*.

Configure the session storage period in the cache

For short-term connection losses to Core Server, you can configure the period during which the session can be reused.

During the specified period, the session is stored in the cache and can be reused when a new server connection is requested. The server availability check is not repeated; the cached check result is used instead.

After the storage period expires, the session is terminated. A new session is created at the next server connection request.

To configure the session storage period in the cache:

In Group Policy Editor

1. Open Group Policy Editor.
2. Open **Computer Configuration → Administrative Templates → Axidian ID → Client Connection**.
3. Enable the **Server connection settings** policy.
4. In the policy properties, set the required value for the **Session object hold period (ms)** parameter. The default value is 180,000 ms (3 minutes).

In registry

1. Open Registry Editor.
2. Open **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2**.
3. For the `SessionHoldPeriodMs` parameter of the *DWORD* type, set the required value. The default value is 180,000 ms (3 minutes).

Password management

ESSO Agent provides for regular password change in applications. Usually the password is changed automatically.

Manual password change by the user can be configured. If manual password change is allowed, then at the next password change the new password value is not generated automatically, but the user is prompted to enter it.

The system behavior when prompting the user for a new password depends on the type of the user account.

If manual password change by the user is enabled, the following window appears at password change:

Enter the new password and confirm it.

- The **Reveal password** button regulates the password entry mode: hidden password or revealed characters.
- The **Generate a random password** button generates a password automatically in accordance with the restrictions set for this application.

  The generated password is automatically pasted into the **Password** field. The **Reveal password** option is then enabled and the **Password confirmation** field is cleared.

A new password entered in the standard mode is checked for compliance with the password complexity criteria set for the application. If the password does not comply with the criteria, an error message appears.

If the ESSO account is of the *Active Directory credentials* type, the following message appears at password change:

Error processing

If an error occurs while the target application is running, ESSO Agent allows you to select an action to process the error. In such cases, the **Error processing** window appears.

In the **Error processing** window, select an action:

- **Repeat**: The operation that resulted in the error is performed again.
- **Close application**: The application that triggered the error is closed. All unsaved data is lost.
- *Close window*: The current window or form of the application is closed. If the main window is closed, the application may be terminated.
- *Ignore*: No action is taken.

After you select an action, the window closes automatically.

Transparent login for a group of users

The Enterprise SSO module allows you to configure transparent login for a specific group of users so that credentials are substituted automatically without using an authenticator.

To configure transparent login without an authenticator for a specific group of users:

1. In the template of the target application, set the `forceReauthentication` parameter to *0*.

   In Management Console, in the application card, on the **General information** tab, the **Authentication when logging in to the application** setting displays the **Log in transparently** value.

2. Make sure that the target application is added to the policy.

3. In the policy settings in Management Console, create an account for the target application with the **Prompt every time** value for the **Authentication when logging in to the application** setting. This is required so that the second factor is mandatory for all users by default.

4. For the users who need transparent login, open the user card, go to the **Applications** tab, open the target application settings, and create an account with the **Log in transparently** value for the **Authentication when logging in to the application** setting.

Users for whom transparent login is configured can select the option without entering an authenticator in the **Select an account** window of ESSO Agent.

Windows Logon

Windows Logon provides users with the following types of access:

- Access to the operating system with the account password.
- Access to the operating system with the available authentication technologies.
- Access to the remote desktop with the available authentication technologies.
- Access to the operating system with a cached authenticator when there is no connection to Core Server.

Windows Logon supports two-factor authentication, certificates, contactless cards, one-time passwords sent by SMS and email, and others.

In Windows Logon, you can authenticate with the following authenticators:

- Email OTP
- Futronic Provider
- Hardware OTP
- Hardware TOTP
- IronLogic Z2 USB Provider
- MFA Provider
- OMNIKEY Provider
- Passcode
- Secured TOTP
- Smart Card Provider
- SMS OTP
- Software OTP
- Storage SMS OTP
- Telegram Provider (in the mode of sending one-time passwords)
- Windows Password
- the Axidian Key mobile application (in the mode of sending one-time passwords and push notifications with login confirmation)

To keep the data secure when the user is away from the workplace, Windows Logon supports manual and automatic workstation locking — when the authentication device is removed or the

screen saver is turned on. Regardless of the locking method, unlocking the workstation always requires the user identity to be confirmed again with an authenticator.

Prerequisites

To use Windows Logon to log in to the system:

1. Install and configure the Windows Logon module.
2. Register the module license in Management Console.
3. Install and configure the authentication providers.
4. If required, install a hardware authentication device.

You can log in to the system with an authenticator and manage authenticators only with the permission granted by the system administrator.

Install

1. Run the installation file located at `Axidian Access <version number>/Axidian Windows Logon/<version number>`.
2. After the component installation is complete, the system must be restarted. In the installation wizard window, click **Yes** to restart immediately, or **No** to do it manually later.

> **NOTE**
> You can deploy Windows Logon on user workstations automatically with Microsoft Group Policy or any other tool that allows you to distribute and install MSI packages on user workstations in bulk (such as Microsoft System Center Configuration Manager).

Configure

You can configure Windows Logon either in Registry Editor or with group policies.

Settings applied with group policies take priority over the settings specified manually in the registry. You can view the group policy settings in the registry at `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\SrvLocator2`.

For more information about the module group policies, see the Windows Logon section.

In registry

1. Open Windows Registry Editor.
2. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2`.
3. Open the properties of the `ServerUrlBase` string parameter.
4. In the **Value** field, specify the URL of your Core Server in the format `http(s)://full_dns_server_name/am/core`.

> **NOTE**
> When using the HTTPS protocol connection, install a client certificate on each Core Server.

In Group Policy Editor

> **NOTE**
> The group policy templates are located in the `Axidian Access <version number>\Misc\ADMX Templates` folder.

1. Add the `AxidianID.ServerUrl.admx` policy to the device with Windows Logon installed.
2. Open Group Policy Editor.
3. Open **Computer Configuration** → **Administrative Templates** → **Axidian ID** → **ClientConnection**.
4. Enable the **Server connection settings** policy.
5. In the **Core Server URL** field, specify the address of your Core Server in the format `http(s)://full_dns_server_name/am/core`.

Additional features

Additional features of Windows Logon:

- Self-registration and management of authenticators with the management utility.
- Generation of a random password for the user in Active Directory.

- The Axidian Access Paste feature, which allows the password to be substituted automatically in a hidden form when a certain key combination is pressed.
- Concurrent operation of the module with RDP Windows Logon.
- Authentication by Axidian Access when starting an application as an administrator.

The first login to the system

If the user has no registered authenticators, the domain password must be used for the first login to the system.

If the user is allowed to register authenticators, the *First login wizard* starts at the login to the system and prompts to register the first authenticator.

You can register the authenticator immediately or later at any convenient time.

- To register an authenticator, in the **Authenticator management** window, select one of the available authentication methods.
- To log in to the system without registering an authenticator, click **Finish**. In this case, the *First login wizard* is displayed at each subsequent login to the system until the first authenticator is registered.

If no authentication providers are detected on the workstation, registering authenticators is not possible. The following error appears: *No authentication provider was found in the system. You must install a provider and configure the secure login to the system again.*

If the user has SMS OTP or Email OTP authentication configured, the login to the system is performed with the configured authenticator, after which a message about password desynchronization appears.

Register the first authenticator

To register the first authenticator:

1. In the wizard window, select the authentication method. The wizard window displays the installed authenticators that are enabled in the Windows Logon properties in the policy card.

2. Perform the required actions following the prompts in the **Authenticator management** window. The window appearance and the actions depend on the selected authentication method.

   If you want to return to the previous page and change the settings or select another login method, click **Back**.

3. After the authenticator is registered, the **Authenticator management** window displays the message *The new authenticator has been successfully registered*. You can add an arbitrary text

comment to the registered authenticator if this action is allowed by the system administrator.

To complete the authenticator registration, click **Save**.

The type of the registered authenticator and the comment to it are displayed in the **Authenticator management** window.

4. If the user is allowed to add several authenticators, you can proceed to register them by clicking **Add a login method**. Depending on the granted permissions, you can also modify, verify, or delete a registered authenticator.

At the next login to the system, the login method with the registered authenticator is available.

Log in to the system with an authenticator

This login method is available only in the following cases:

- You are a licensed user of Axidian Access.
- Your account is allowed to register authenticators.
- You have already registered at least one authenticator.

To log in to the system:

1. After the operating system boots, select the required user.
2. The **Windows login** window that opens displays the name of the user who logged in to the system last and the login method they used.

   If you want to switch the user, click **Cancel** and select the account of the required user.

   If you want to select another login method, click **Switch the login method** and select one of the login methods corresponding to a registered authenticator.
3. After selecting the login method, follow the prompts in the authentication window. The appearance of the authentication window and the required actions depend on the selected authenticator type.

Password desynchronization

In a number of cases, password desynchronization may occur:

- If the domain password was reset by the system administrator while the user has at least one registered authenticator.
- If no Core Server is available at the moment of the password change request, but at least one domain controller is available. In this case, the following message appears: *Server communication error. Server not found. If you continue, password desynchronization occurs.*
- At the first login to the system, if the user has SMS OTP or Email OTP authentication configured.

This happens when the new password cannot be synchronized with the Core Server database.

To synchronize the password:

1. In the window with the desynchronization warning, click **OK**.
2. Enter the new password.

   If the desynchronization occurred because Core Server was unavailable, perform these actions when at least one server is available.

The current password from Active Directory is synchronized with the password in the Core Server database. After successful synchronization, the login to the system is performed.

Access to the remote desktop

To access the remote desktop with an authenticator, the following is required:

- The standard Windows *Remote Desktop Connection* utility (mstsc.exe) on the device the connection is made from (the terminal client).
- The Windows Logon module installed on the device that requires remote access (the terminal server).
- The corresponding authentication provider installed on the device that requires remote access.
- When using Hardware OTP Provider, a hardware authentication device connected to the device the connection is made from.
- Network Level Authentication disabled on the device that requires access.

How to disable Network Level Authentication

In registry

1. Open Registry Editor.
2. Open **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services**.
3. For the `UserAuthentication` parameter of the *DWORD* type, set the value *0*.
4. For the `MinEncryptionLevel` parameter of the *DWORD* type, set the value *3*.
5. For the `SecurityLayer` parameter of the *DWORD* type, set the value *0*.

In Group Policy Editor

1. Open Group Policy Editor.
2. Open **Computer Configuration** → **Administrative Templates** → **Windows Components** → **Remote Desktop Services** → **Remote Desktop Session Host** → **Security**.
3. Disable the **Require user authentication for remote connections by using Network Level Authentication** policy.
4. Enable the **Set client connection encryption level** policy.
5. In the policy parameters, for the **Encryption Level** setting, select the **High Level** value.
6. Enable the **Require use of specific security layer for remote (RDP) connections** policy.
7. In the policy parameters, for the **Security Layer** setting, select the **RDP** value.

⚠ **WARNING**

If **Network Level Authentication** (NLA) is disabled, the security level decreases. Therefore, we recommend that you use the RDP Windows Logon module to connect to the remote desktop.

You can also configure the concurrent operation of Windows Logon and RDP Windows Logon.

To connect to the remote desktop with an authenticator:

1. Start **Remote Desktop Connection**.
2. Enter the name or address of the device you need to access and click **Connect**.
3. In the **Windows Security** window, enter the account password and wait for the connection.
4. In the **Windows login** window that appears, select the account and the login method (the authenticator type) and authenticate.

> ⓘ **NOTE**
> If random password generation is not enabled for the account, you can access the remote desktop with the domain password.

To avoid being prompted for the username and password when connecting to the remote desktop, you need to change the system settings. For more information, see the article Disabling the login and password request on the terminal client side.

Login when the network is unavailable

You can log in to the system with an authenticator even without a physical network connection. Logging in with a cached authenticator is similar to logging in with an authenticator when the network is available.

To configure login when the network is unavailable:

1. Configure session caching with group policies or in the registry.
2. Enable caching and configure the authenticator validity period in Management Console.
3. Configure the session storage period in the cache.

Prerequisites

Login when the network is unavailable is possible in the following cases:

- Your account has been granted permission to cache authenticators.
- Authenticators were cached at the first login to the system with an authenticator.
- A login with a cached authenticator was performed while the network was available.

Configure session caching

When there is no connection to Core Server, users can authenticate in Windows Logon in the offline mode. For this, a session cached on the user computer is used.

To configure login with a cached session, use the **Session public key settings** policy.

In registry

1. Open Registry Editor on the computer.
2. Open **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\AM\CacheServer**.
3. Create the **SessionPublicKey** string parameter.
4. Set the parameter to the value of the public key generated during the Core Server installation. You can find the public key value in the `SessionPublicKey.pub` file or in the Core Server `Web.config` configuration file, in the `sessionEncryptionSettings` section, in the `publicKey` parameter.

5. Go to **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID**.
6. Create the **ServerAvailability** section in it.
7. In this section, create three parameters of the *DWORD* type:
   - `IsEnabled` with the value *1*
   - `CheckResultTtlMs` with the required check result lifetime value in ms
   - `CheckWaitTimeoutMs` with the required check result waiting timeout value in ms

In Group Policy Editor

1. Add the `AxidianID.SessionPublicKey.admx` policy file from the `Axidian Access <version number>\Misc\ADMX Templates` folder to the computer with Core Server and Windows Logon installed.
2. Open Group Policy Editor.
3. Open **Computer Configuration** → **Administrative Templates** → **Axidian ID** → **Cache server**.
4. Enable the **Session public key settings** policy.
5. In the policy properties, in the **Session public key** field, enter the value of the public key generated during the Core Server installation. You can find the public key value in the `SessionPublicKey.pub` file or in the Core Server `Web.config` configuration file, in the `sessionEncryptionSettings` section, in the `publicKey` parameter.
6. Go to **Computer Configuration** → **Administrative Templates** → **Axidian ID** → **Client Connection**.
7. Enable the **Axidian EA\ESSO server availability check** policy.
8. In the policy parameters, set the values for the **Check result waiting timeout (ms)** and **Check result lifetime (ms)** settings.

Enable caching and configure the authenticator validity period

You can enable caching and set the validity period of cached authenticators in the application settings of a policy or a user in Management Console. If the validity periods are not specified, caching is valid indefinitely.

If the **Number of days** parameter is set for cached authenticators, their validity period is counted from the time of the last login to Windows Logon with the network available, using any authenticator except Windows Password.

For example, if the period is set to 10 days and the login was performed at 10:00 on 12.04.2025, the cached authenticators are valid from 10:00 on 12.04.2025 until 10:00 on 22.04.2025.

If the validity period of cached authenticators has expired and Core Server is unavailable, the following message appears when you try to log in: *Login error. The user data caching period has expired*.

Configure the session storage period in the cache

For short-term connection losses to Core Server, you can configure the period during which the session can be reused.

During the specified period, the session is stored in the cache and can be reused when a new server connection is requested. The server availability check is not repeated; the cached check result is used instead.

After the storage period expires, the session is terminated. A new session is created at the next server connection request.

To configure the session storage period in the cache:

In registry

1. Open Registry Editor.
2. Open **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2**.
3. For the `SessionHoldPeriodMs` parameter of the *DWORD* type, set the required value. The default value is 180,000 ms (3 minutes).

In Group Policy Editor

1. Open Group Policy Editor.
2. Open **Computer Configuration → Administrative Templates → Axidian ID → Client Connection**.
3. Enable the **Server connection settings** policy.

4. In the policy properties, set the required value for the **Session object hold period (ms)** parameter. The default value is 180,000 ms (3 minutes).

Optional settings

With Windows Logon, you can:

Enable random password generation

When using the Windows Logon module, you can set up random password generation for the user in Active Directory. For this setting to work, disable the **User cannot change password** and **Password never expires** options in the user properties in Active Directory.

A random password for the account is generated when the current password expires. If a random password was generated for the account, the next login to the system is possible only with an authenticator.

To enable random password generation:

1. In the Management Console sidebar, go to the **Policies** section.
2. Open the required policy.
3. On the **Applications** tab, click **Windows Logon**.
4. For the **Active Directory account password** setting, select **Generate random**.
5. Click **Save**.

The passwords for all users in this policy are changed unless this contradicts their properties in Active Directory.

The Management Console event log displays event *1091* **The user password has been successfully changed automatically**. Users can log in to the system only with the added authenticators, without using the domain password.

If the user has no registered authenticators, at the first login to the system, after entering the domain password, the authenticator management utility opens and prompts to register the **first authenticator**.

After the authenticator is registered, the user password is generated and changed.

If the domain password was changed by the administrator, a message about password desynchronization appears at the login to the system. Click **OK** and enter the password set by the administrator. After that, the password is changed automatically.

> **INFORMATION**
> If authenticator caching is allowed for the account, the data is saved to the local computer memory at the subsequent login with an authenticator. Then, when there is no connection to Core Server, you can log in to the system with the cached authenticator data.

If the password change requirement is enabled for the user

```
If the domain password change requirement is enabled in the user properties in Active
Directory and random password generation is allowed, a random password is generated
at the next authentication with Axidian Access technology. A message about the
successful automatic password change is displayed.
```

Configure automatic substitution of the hidden password with Axidian Access Paste

When a password must be entered to log in to an application, you can do it securely with the Axidian Access Paste feature. It allows the hidden password to be substituted automatically into the input field with a certain key combination. By default, this is *[CTRL] + [ALT] + [V]*.

You can enable and disable the Axidian Access Paste feature in the Paste Tool application. To do this:

1. Right-click the Paste Tool icon in the Windows notification area.
2. Select the required item of the context menu:

- **Enable Paste** to enable or disable the feature. The feature is enabled by default.
- **Run at startup** so that Paste Tool starts when Windows boots. This setting is enabled by default.
- **Exit**.

To substitute the password in a hidden form:

1. In the application window, place the cursor in the password input field.
2. Press the key combination for password substitution.
3. The **Authentication** window opens. Confirm your identity in any available way.

After successful authentication, the hidden password is displayed in the input field.

Configure concurrent operation with RDP Windows Logon

If the Windows Logon and RDP Windows Logon modules are installed on the same device, configure the policy for Windows Logon.

In registry

1. Go to the device with the RDP Windows Logon and Windows Logon modules installed.
2. Open Registry Editor.
3. Open **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Logon for Windows**.
4. Create a parameter of the *DWORD* type named `CredProvFilter` and set it to the value *2*.

In Group Policy Editor

1. Go to the device with the RDP Windows Logon and Windows Logon modules installed.
2. Open Group Policy Editor.
3. Open **Computer Configuration** → **Administrative Templates** → **Axidian ID** → **Windows Logon**.
4. Enable the **Credential Provider settings** policy.
5. For the **Display of login methods** parameter, select the **All except the password** value.

Use authentication when running as an administrator

On devices running Windows 7, the Windows Logon module sends an authentication request when the **Run as administrator** command is used. After this command is run, the account selection dialog is displayed, then the Axidian Access **Authentication** window. Depending on the operating system version, the account selection window may look different.

Authenticator management utility

> ⚠ **IMPORTANT**
>
> The authenticator management utility works only in scenarios with the Enterprise SSO and Windows Logon modules.
>
> The utility is installed automatically when these modules are installed.

You can manage authenticators with the Axidian Access Authenticator management application. This application allows you to add new authenticators and edit, verify, and delete the existing ones.

> ⓘ **NOTE**
>
> The authenticator management capabilities are available if the corresponding permission is granted by the system administrator.

To proceed to authenticator management, perform the following actions:

1. Open the Axidian Access Authenticator management application (**Start→Axidian Access→Axidian Access Authenticator management**).
2. In the **Authentication** window, select your account and the login method (the authenticator type). By default, the last used login method is selected automatically.

   > ⓘ **INFORMATION**
   >
   > To log in to the Authenticator management application, use a previously registered authenticator. If you use the password to access the application, the functions of modifying, deleting, and adding authenticators are not available.

3. Perform the required actions (depending on the selected login method).

   > ⓘ **NOTE**
   >
   > The capabilities of adding, editing, and deleting authenticators are available after authorization with an authenticator. The capability of verifying authenticators is available both after authorization with an authenticator and after authorization with the password.

4. After successful authentication, the **Authenticator management** window is displayed.

Add an authenticator

> ⚠ **IMPORTANT**
> Before registering authenticators, you must create a policy and apply it to the target users.

The number of authenticators you can add is set by the system administrator. When the maximum number of authenticators is reached, the **Add a login method** button in the **Authenticator management** window becomes inactive.

To add an authenticator, perform the following actions:

1. In the **Authenticator management** window, click **Add a login method**.
2. Select the authentication technology.
3. Follow the instructions for registering the authenticator. The window appearance and the prompt text depend on the selected authenticator type.

After the authenticator is registered successfully, the **Authenticator management** window displays the message *The new authenticator has been successfully registered*.

4. You can add an arbitrary text comment to the registered authenticator (if this action is allowed by the system administrator). To complete the authenticator registration, click **Save**.

The type of the registered authenticator and the comment to it are displayed in the **Authenticator management** window. If your account is allowed to add several authenticators, you can proceed to register them by clicking **Add an authenticator**.

Verify an authenticator

To verify an authenticator, perform the following actions:

1. In the **Authenticator management** window, select the authenticator and click **Verify**.
2. The **Authenticator management** window then displays the instructions for registering the authenticator. The window appearance and the prompt text depend on the selected authenticator type. Perform the required actions.
3. After the verification is complete, the **Authenticator management** window displays a message with the result.
    - If the authenticators match:

- If the authenticators do not match:

4. If the authenticators do not match, click **Back** and repeat the verification procedure or register the authenticator again.

Edit an authenticator

> ⚠ **IMPORTANT**
>
> Available actions:
>
> - Edit (register again) the authenticator.
> - Change only the comment to the authenticator (if this action is allowed by the system administrator).

To edit an authenticator, perform the following actions:

1. In the **Authenticator management** window, select the authenticator and click **Edit**.
2. Perform one of the following actions:
    - To edit only the comment to the authenticator, enter the comment in the corresponding field and click **Save**.

- To proceed to editing the authenticator itself, click **Set again**. The window then displays the instructions for registering the authenticator. The window appearance and the prompt text depend on the selected authenticator type. Perform the required actions.

3. After the authenticator is registered successfully, click **Save**.

Delete an authenticator

To delete an authenticator, in the **Authenticator management** window, select the authenticator and click **Delete**. To confirm the action, click **Yes**.

If a random password was generated for your account, when you try to delete the only authenticator you have, the system displays a warning dialog:

> **TIP**
> If you have deleted all authenticators and cannot log in to the system with the password, contact the system administrator. After the password is reset, you can log in to the system again and register new authenticators.

After all authenticators are deleted, the login to the system can be performed only with the password. If a random password was generated for your account, the login to the system becomes impossible.

Authentication providers

Windows Password

Authentication with the Windows password

Axidian Key

A one-time password and push notifications in the Axidian Key application

Software OTP

A one-time password in addition to the login and password

Passcode

Authentication with a user-defined password

HOTP

Authentication with a Hardware OTP from eToken PASS

Email OTP

One-time passwords via email

MFA Provider

Multi-factor authentication

SMS OTP

2 items

Storage SMS OTP

Authentication with phone numbers from a Microsoft SQL database

OMNIKEY

Authentication with HID OMNIKEY smart cards

Futronic

Authentication with a biometric scanner

Telegram

A one-time password and push notifications in the Telegram application

Secured TOTP

A one-time password based on the device ID

IronLogic Z2 USB

Authentication with the IronLogic Z2 USB reader

Hardware TOTP

Authentication with a TOTP from eToken PASS

Smart Card

Authentication with smart cards and USB keys

Migrate provider settings

Transferring provider settings to the Management Console

Windows Password

Windows Password can be used for authentication in the following modules:

- Identity Provider
- Windows Logon

Provider ID
{CF189AF5-01C5-469D-A859-A8F2F41ED153}

Install the provider

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian Windows Password Provider/<version number>` and follow the steps of the installation wizard.
2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.
3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Credentials verification

When authenticating through Windows Password Provider, you can configure the credentials verification on the domain controller. To do this:

1. In the **Management Console**, go to **Configuration** → **Authenticators** and select **Windows Password** from the list.

2. In the **General settings** section, select the preferred credentials verification method:

    - **Logon User** (the default value) — used only for Windows on machines inside the domain, or on machines from another domain if a trust relationship is configured between the domains. If Logon User is selected but cannot be used, the Ldap Bind password verification method is used instead.
    - **Ldap Bind** — used if Core Server on Windows is deployed outside the domain, or if Core Server is deployed on Linux.

> **NOTE**
>
> If you select Ldap Bind, a user who tries to sign in with an expired Windows password does not receive the *The password has expired* notification.

Axidian Key

Authentication with push notifications has the following requirements:

- In the network infrastructure:
  - Axidian Key Server is installed. This server sends notifications to the user smartphone through the Google API services.
  - Axidian Key authentication provider is installed. Installing the provider allows you to use this authentication technology in the target scenarios.
- On the client side:
  - The Axidian Key application is installed. The application is required to receive authentication notifications in the target applications through the Axidian Access components.

Authentication with push notifications is impossible without Axidian Key Server, Axidian Key Provider, and the Axidian Key application.

Axidian Key Provider can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- IIS Extension (only in the mode of sending push notifications with sign-in confirmation)
- NPS RADIUS Extension (only in the mode of sending push notifications with sign-in confirmation)
- RDP Windows Logon (only in the mode of sending push notifications with sign-in confirmation)
- Windows Logon

Provider ID
{DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}

Install Axidian Key Provider

To install the provider:

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian Key Provider` and follow the steps of the installation wizard. Install the provider both on the computer with Core Server installed and on the client computer.

   > ⚠ **IMPORTANT**
   > If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.
   >
   > If the provider is used in client scenarios with Windows Logon and ESSO Agent, install the provider from the *Client* folder on the client machines.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Configure Axidian Key Provider in the Management Console

To configure Axidian Key in the Management Console, go to **Configuration** → **Authenticators** and select the Axidian Key authenticator.

Access rights

To grant or revoke the rights to use the authenticator:

1. In the **General settings** section, configure the following:
   - In the **Forbid to use** setting, specify whether the user can use or register the selected authenticator.
2. In the **Actions available to the user** section, configure the following:
   - In the **Register new authenticators** setting, specify whether the user can register new authenticators.
   - In the **Edit existing authenticators** setting, specify whether the user can modify the authenticators that are already registered.
   - In the **Delete existing authenticators** setting, specify whether the user can delete the authenticators that are already registered.

- In the **Allow editing the authenticator comment** setting, specify when the user can edit the comments of the authenticators that are already registered.

Block the authenticator

To block the authenticator, configure the following in the **Authenticator blocking settings** section:

1. Allow or block the use of Axidian Key in case of a series of failed authentication attempts.
2. In the **Number of authentication attempts before blocking** field, specify how many times the user can fail authentication before the authentication method is blocked.
3. In the **Blocking counter reset** field, specify how many minutes must pass after a failed authentication attempt before the counter is reset.
4. In the **Timeout before the sign-in method is unblocked** field, specify after how many minutes the blocked authentication method becomes available again.

Mobile application settings

To configure the Axidian Key mobile application, configure the following in the **Mobile application settings** section:

1. In the **Displayed server name** field, specify the name of the Axidian Key server that is displayed in the mobile application.
2. In the **Push confirmation method** setting, select how the user is asked to confirm the sign-in:
   - **No confirmation required** — the user is not asked for additional authentication on the device.
   - **Confirmation with biometrics** — the user is asked for additional authentication on the device with biometrics.
   - **Confirmation with any method available on the device** — the user is asked for additional authentication on the device with any method available on it.
3. In the **Display the one-time password in the mobile application for new authenticators** setting, specify whether the one-time code is displayed in the authenticator card in the application. By default, the one-time code is hidden.
4. In the **Confirmation method for displaying the OTP** setting, select how the user confirms the right to view the one-time codes:
   - **No confirmation required** — the user is not asked for additional authentication on the device.
   - **Confirmation with biometrics** — the user is asked for additional authentication on the device with biometrics.

    - **Confirmation with any method available on the device** — the user is asked for additional authentication on the device with any method available on it.
5. In the **OTP display time** field, specify how long the one-time password remains valid. The value is specified in seconds.
6. In the **Axidian Key operating mode** setting, select the sign-in confirmation method used in the application:
    - **Push** — the user receives push notifications with the option to reject or confirm the sign-in.
    - **OTP** — instead of push notifications, the user receives a one-time code.

Server settings

To configure the server that the Axidian Key application connects to, configure the following in the **Server settings** section:

1. In the **Axidian Key Server URL** field, specify the address at which the Axidian Key server is available from the Axidian Access machine.
2. In the **Axidian Key Server trusted ID** field, specify an arbitrary unique identifier. This identifier is required to confirm the authenticator deletion. The identifier value must match the value specified in the Axidian Key Server configuration file (the `trustedClients` tag).

One-time password settings

To configure the one-time password generation, configure the following in the **OTP settings** section:

1. In the **OTP refresh period** field, specify how long the one-time password remains valid.
2. In the **Comparison window** setting, specify how many refresh periods must pass before the one-time password is considered invalid.
3. In the **User name format** setting, select the format in which the user name is displayed in the application. The default value is *Name*.
4. In the **OTP code generation algorithm** setting, select the algorithm used to generate the one-time password.
5. In the **Number of digits in the OTP code** setting, select how many digits the one-time password contains.

Register the Axidian Key authenticator via a link

Registering the Axidian Key authenticator via a link from an email requires the following conditions:

- Email OTP Provider is installed and configured.

- The user email is filled in the Active Directory catalog.

The link is sent to the email address specified for the user in the Active Directory catalog.

To register Axidian Key via a link:

1. Go to **Configuration** → **Authenticators**, and in the **Registration settings** section, configure the following:
    1. In the **Authenticator registration method** setting, select the registration via a link from an email.
    2. Then specify the following:
        - In the **Notification subject** field, specify the subject of the registration email.
        - In the **Message** field, specify the email body.
        - In the **Link text** field, specify the text of the link that the user follows to register the authenticator.
        - If required, specify the server address used to form the link. If you leave the field empty, the default link is used, depending on the Axidian Access locale.
2. Then go to the **Users** section, select a user, and open the **Authenticators** tab.
3. Click **Register** and select Axidian Key from the list.

After that, the user receives an email with a link to the authenticator registration. As soon as the email is sent, the Axidian Key authenticator with the *Pending* status is added in the Management Console and in the User Console. Following the link, the user goes to the Axidian Key application, where the authenticator registration starts.

After the user has completed the registration in the application, the administrator must click the status refresh icon in the Management Console and then click **Save**. The *Pending* status changes to *Active*, and the authenticator is ready to use.

If the user encounters errors during the Axidian Key registration, the *Pending* status changes to *Error* after the status is refreshed in the Management Console. Delete the authenticator and repeat the registration procedure.

Register the Axidian Key authenticator with a QR code

To register Axidian Key with a QR code:

1. In the **Actions available to the user** section, allow the user to register new authenticators.

2. In the **Authenticator registration method** setting, select the registration with a QR code.

Axidian Key appears in the list of available authenticators for the user in the User Console.

The user must do the following:

1. Click the gear icon.
2. Click **Register**.
3. Open the Axidian Key application and click **+**.
4. Scan the QR code that appears.

Configuring the registration link and the Axidian Key download link

The link to the authenticator registration and to the Axidian Key application download can be defined manually. To do this:

1. Go to **Configuration** → **Authenticators**, to the **Registration settings** section.
2. In the **Server address used to form the authenticator registration link (deeplink)** setting, specify the registration link that is sent by email.
3. In the **Additional settings** section, specify the link to download Axidian Key.

Spam protection

The spam protection mechanism is based on calculating the percentage of successful authentications relative to all sent messages over a specified time interval. The calculation process starts only if the number of sent messages exceeds the number that you have specified in the Evaluation Window setting.

When a spam attack is detected, further message sending is blocked for a specified period of time, and a *"Potential spam attack detected"* error occurs when attempting to log in.

Messages can be sent again either after the specified period expires or when a certain percentage of successful authentications is reached.

To configure spam protection, perform the following actions:

1. In Management Console, in the **Configuration→Authenticators** section, select an authenticator.
2. In the **Spam Protection Settings** section, configure the following settings:

- Enable or disable spam protection.
- In the **Authentication attempts evaluation window** field, specify the time period during which the percentage of successful login attempts is calculated.
- In the **Authentication attempts threshold window** field, specify how many login attempts must be made during the time specified in the authentication attempts evaluation window.
- In the **Percentage of successful authentication attempts** field, specify the minimum percentage of successful logins relative to all sent messages.

Example

The setting is enabled with the following values:

- **Authentication attempts evaluation window** — 600
- **Authentication attempts threshold window** — 20
- **Percentage of successful authentication attempts** — 85

Spam protection is activated if 21 login attempts occur (21 messages sent). The authenticator will be blocked for 600 seconds.

The authenticator will be unblocked in one of the following cases:

- the percentage of successful logins (user successfully entered the one-time password from the message) reaches 85;
- 600 seconds have passed since the blocking.

INFORMATION

Log server events:

- 2090: Potential spam attack detected. Message sending suspended.
- 1118: Message sending to users resumed.

Configure the trusted identifier for Axidian Key Server

By default, when an authenticator is deleted in the User Console, in the authenticator management tool, or in the administrator console, the authenticator is removed from the system without

notifications and continues to be displayed in the application.

To configure push notifications about the authenticator deletion and the deletion in the application:

1. Open the Axidian Key Server configuration file `Web.config` from the `C:\inetpub\wwwroot\axidiankey` folder.

2. Add the following tags to the file after the `appSettings` block:

   > **NOTE**
   > In the `id` parameter of the `add` tag, enter the unique identifier specified in the **Axidian Key Server trusted ID** setting in the Management Console.

   Example

   ```
   <trustedClientsSettings>
       <trustedClients>
         <add id="the unique identifier set when configuring Axidian Key Server" />
       </trustedClients>
     </trustedClientsSettings>
   ```

3. After the parameters are applied, the user receives notifications about the key deletion.

Configure the Axidian Key application on a smartphone

The Axidian Key application is available for smartphones with iOS 13.0 and higher, and Android 7.0 and higher.

The Axidian Key application is guaranteed to work correctly only if it is downloaded from the official application stores. To download or update the application, use the following links:

- App Store
- Google Play

> **IMPORTANT**
> To update the application, download the update from the same store where you installed the previous version.

A note for Android device owners

For push notifications to work correctly on Android devices, make sure that the following mandatory requirements are met:

- You have allowed the application to send notifications.
- On devices with Android 9 and higher, the power saving mode is disabled.

Note: push notifications may not work correctly on all Android modifications.

1. Open the application. In the **Authenticators** panel, click **+** to add an authenticator.
2. Allow the application to access the camera and scan the QR code.
3. Confirm the authenticator registration by clicking **Confirm**.
4. After a successful registration, the authenticator is displayed in the main application window.

To delete an authenticator, select it and click **Delete authenticator**.

ADDITIONAL SETTINGS

For machines with the Windows Logon and ESSO Agent components installed, configure the authentication parameters with group policies.

Software OTP

With Software OTP Provider, you can implement software-based two-factor authentication. The authenticator is a one-time password that the user must provide in addition to their login and password to get access to an application.

The one-time password is generated autonomously on a mobile device (a phone, a smartphone, or a tablet) with a dedicated application (for example, Axidian Key or Google Authenticator). The one-time password is generated based on two parameters: the secret key set at the authenticator registration stage, and the current time.

The authentication technology is based on a system where, for a given user secret key, only one valid one-time password exists at any given moment. Therefore, knowing the secret key, the server can verify the one-time password provided by the user. For the technology to work correctly, the time on the mobile device and on the authentication server must match (a deviation is allowed, and its value can be adjusted by the administrator).

Software OTP can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- IIS Extension
- NPS RADIUS Extension
- RDP Windows Logon
- Windows Logon

Provider ID
{0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}

Provider ID for NPS RADIUS Extension
{B772829C-4076-482B-B9BD-53B55EA1A302}

Prerequisites

The provider requires Bsp Broker installed on the Axidian Access server.

Files for Bsp Broker are located at `Axidian Access <version number>/Axidian Providers/Axidian Bsp Broker/<version number>`.

- `AuthProviders.BspBroker-<version number>.x64.en-us.msi` — the installation package of Bsp Broker for 64-bit operating systems.
- `AuthProviders.BspBroker-<version number>.x86.en-us.msi` — the installation package of Bsp Broker for 32-bit operating systems.

Install the provider

1. Run the `AuthProviders.SoftwareTOTP.Provider-<version number>.<bitness>.en-us.msi` installation file located at `Axidian Access <version number>\Axidian Providers\Axidian Software OTP Provider\<Version number>` and follow the steps of the installation wizard.

   > ⚠ **IMPORTANT**
   > If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Policy template files are located at `Axidian Access <version number>\Misc\ADMX Templates`.

Forced authenticator verification

You can configure forced authenticator verification for the authenticator registration.

1. Go to **Configuration** → **Authenticators** and select **Software TOTP** from the list.
2. In the **General settings** section, enable the forced verification. An additional confirmation window for entering the authentication data appears during registration.
3. Enter the authentication data you have received and click **Confirm**.

Passcode

Passcode Provider is designed for user authentication with a password that users set themselves, and for joint use with the Axidian Access products.

Passcode can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- IIS Extension
- NPS RADIUS Extension
- RDP Windows Logon
- Windows Logon

Provider ID
`{F696F05D-5466-42b4-BF52-21BEE1CB9529}`

Prerequisites

The provider requires Bsp Broker installed on the Axidian Access server.

Files for Bsp Broker are located at `Axidian Access <version number>\Axidian Providers\Axidian Bsp Broker\<Version number>`.

- *AuthProviders.BspBroker-<version number>.x64.en-us.msi* — the installation package of Bsp Broker for 64-bit operating systems.
- *AuthProviders.BspBroker-<version number>.x86.en-us.msi* — the installation package of Bsp Broker for 32-bit operating systems.

Install the provider

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian Passcode Provider/<version number>` and follow the steps of the installation wizard.

> ⚠ **IMPORTANT**
>
> If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

> ⓘ **ADDITIONAL SETTINGS**
>
> To configure the password requirements, go to the group policies section for Passcode.

HOTP

The eToken PASS standalone one-time password generator can be used for authentication in any applications and services that support the RADIUS authentication protocol: VPN, Microsoft ISA, Microsoft IIS, Outlook Web Access, and others.

eToken PASS implements the One-Time Password (OTP) generation algorithm. This algorithm is based on the HMAC algorithm and the SHA-1 hash function. Two input parameters are used to calculate the OTP value — the secret key (the initial value for the generator) and the current counter value (the number of generation cycles required). The initial value is stored both in the device itself and on the server in Axidian Access. The counter in the device is incremented at every OTP generation, and on the server — at every successful OTP authentication.

Hardware OTP can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- IIS Extension
- NPS RADIUS Extension
- RDP Windows Logon
- Windows Logon

Provider ID
`{AD3FBA95-AE99-4773-93A3-6530A29C7556}`

Install the provider

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian HOTP Provider/Server/<version number>` and follow the steps of the installation wizard.
2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.
3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Add a device

> **INFORMATION**
>
> A device can be registered for one user only.

To add a device:

1. Open the Management Console.
2. Go to the **Devices** tab.
3. Click **Add** or **Add from file**.

> **NOTE**
>
> You can add a device either by selecting a file with the device parameters (**Add from file**) or by specifying the parameters manually (**Add**).

Add from a file

1. Click **Add from file**.
2. Select the XML file with the device parameters and click **Add**.

An example of the device configuration

```
<Tokens xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Token serial="000200071927">
            <CaseModel>5</CaseModel>
            <Model>109</Model>
            <ProductionDate>11/4/2008</ProductionDate>
            <ProductName>Aladdin OTPO v1.0</ProductName>
        <Applications>
            <Application ConnectorID="{a61c4073-2fc8-4170-99d1-9f5b70a2cec6}">
            <Seed>884f20ce4b2c406e0b6199338990bb6cc3fabac403eaa7f8</Seed>
            <MovingFactor>1</MovingFactor>
            </Application>
        </Applications>
    </Token>
</Tokens>
```

Add manually

1. Click **Add**.

2. In the **Serial number** field, specify the device serial number.
3. In the **Secret key** field, specify the device seed.
4. The **Comment** field is an optional parameter.
5. Click **Add**.

Edit a device

To edit a device:

1. Open the Management Console.
2. Go to the **Devices** tab.
3. Click the device serial number.

> **INFORMATION**
> If required, you can search using the filters at the top of the page.

4. In the editing window, you can change the comment, block the device, or synchronize it.
5. After you make the changes, click **Save**.

Delete a device

To delete a device:

1. Open the Management Console.
2. Go to the **Devices** tab.
3. Select the device and click **Delete**.

> **INFORMATION**
> If required, you can search using the filters at the top of the page.

Delete from a file

1. To delete a device using the information from a file, click **Delete from file**.
2. Select the device configuration file.

3. Click **Delete**.

Synchronize a device

To synchronize a device:

1. Open the Management Console.
2. Go to the **Devices** tab.
3. Click the device serial number.
4. Select **Synchronize OTP**.
5. In the **One-time password 1** and **One-time password 2** fields, enter the passwords from the device and click **Synchronize**.

Email OTP

The Email OTP Provider component is designed for user authentication with one-time passwords delivered to users by email.

A one-time password is a set of random characters (digits, special characters, and Latin letters). The password is generated on the Axidian Access server, then it is transmitted to the email delivery service, which forwards it to the user as an email message. Data is transmitted through the SMTP (Simple Mail Transfer Protocol) protocol.

Email OTP can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- IIS Extension
- NPS RADIUS Extension
- RDP Windows Logon
- Windows Logon

Provider ID
`{093F612B-727E-44E7-9C95-095F07CBB94B}`

Prerequisites

To use Email OTP Provider, an email server is required. This server must be accessible from every Axidian Access server where Email OTP Provider is to be installed.

To use the authenticator, the user must have an email address specified in the `mail` attribute. Otherwise, the authenticator is not available for use.

Authenticator registration in the User Console is not required.

Install the provider

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian Email OTP Provider/Server/<version number>` and follow the steps of the installation wizard.
2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.
3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Configure the email attribute

To change the default attribute, add the following parameters to the server configuration file `C:\inetpub\wwwroot\am\core\Web.config`:

- Add the `userMapRules` tag inside the `adUserCatalogProvider` tag.
- Add the `adObjectMapRule` tag inside the `userMapRules` tag with the following parameters:
  - `attribute="Email"` — specifies the parameter that is changed
  - `adAttribute="otherMailbox"` — specifies the Active Directory attribute that the value is taken from
- Add the `objectTypeSettings` tag.
- Add the `objectSetting` tag with the `category="person" class="user"` parameters.

Example

```
<adUserCatalogProvider id="userId" serverName="axidian.local"
containerPath="DC=axidian,DC=local" userName="userAdmin" password="Q1q2E3e4">
                <userMapRules>
                        <adObjectMapRule attribute="Email" adAttribute="otherMailbox"/>
                    <objectTypeSettings>
                        <objectSetting category="person" class="user"></objectSetting>
                    </objectTypeSettings>
                </userMapRules>
        </adUserCatalogProvider>
```

SMTP server settings

To work with the SMTP server, configure the group policy.

Block the authenticator

To block the authenticator:

1. In the Management Console, go to **Configuration** → **Authenticators** and select the Email OTP authenticator from the list.
2. In the **Authenticator blocking settings** section, configure the following:
    - Allow or block the use of Email OTP in case of a series of failed authentication attempts.
    - In the **Number of authentication attempts before blocking** field, specify how many times the user can fail authentication before the authentication method is blocked.
    - In the **Blocking counter reset** field, specify how many minutes must pass after a failed authentication attempt before the counter is reset.
    - In the **Timeout before the sign-in method is unblocked** field, specify after how many minutes the blocked authentication method becomes available again.

One-time password settings

The settings are applied to the Axidian Access servers and allow you to define the length of the one-time password and the character groups it contains.

To configure the one-time password generation:

1. In the Management Console, go to **Configuration** → **Authenticators** and select the Email OTP authenticator from the list.
2. In the **One-time password generation settings** section, configure the following:
    - In the **One-time password length** field, specify how many characters the one-time password contains.
    - In the **Digits** setting, specify whether the one-time password contains digits.
    - In the **Lowercase Latin letters** setting, specify whether the one-time password contains lowercase Latin letters.
    - In the **Uppercase Latin letters** setting, specify whether the one-time password contains uppercase Latin letters.
    - In the **Special characters** setting, specify whether the one-time password contains special characters.

Spam protection

The spam protection mechanism is based on calculating the percentage of successful authentications relative to all sent messages over a specified time interval. The calculation process starts only if the number of sent messages exceeds the number that you have specified in the Evaluation Window setting.

When a spam attack is detected, further message sending is blocked for a specified period of time, and a *"Potential spam attack detected"* error occurs when attempting to log in.

Messages can be sent again either after the specified period expires or when a certain percentage of successful authentications is reached.

To configure spam protection, perform the following actions:

1. In Management Console, in the **Configuration→Authenticators** section, select an authenticator.
2. In the **Spam Protection Settings** section, configure the following settings:
    - Enable or disable spam protection.
    - In the **Authentication attempts evaluation window** field, specify the time period during which the percentage of successful login attempts is calculated.
    - In the **Authentication attempts threshold window** field, specify how many login attempts must be made during the time specified in the authentication attempts evaluation window.
    - In the **Percentage of successful authentication attempts** field, specify the minimum percentage of successful logins relative to all sent messages.

Example

The setting is enabled with the following values:

- **Authentication attempts evaluation window** — 600
- **Authentication attempts threshold window** — 20
- **Percentage of successful authentication attempts** — 85

Spam protection is activated if 21 login attempts occur (21 messages sent). The authenticator will be blocked for 600 seconds.

The authenticator will be unblocked in one of the following cases:

- the percentage of successful logins (user successfully entered the one-time password from the message) reaches 85;
- 600 seconds have passed since the blocking.

INFORMATION

Log server events:

- 2090: Potential spam attack detected. Message sending suspended.
- 1118: Message sending to users resumed.

MFA Provider

MFA Provider allows you to define the sequence of providers in a multi-factor authentication chain.

MFA Provider can be used for authentication in the following modules:

- Enterprise SSO
- NPS RADIUS Extension
- Windows Logon

Provider ID
`{070719BA-EB57-4EA8-BB4D-D15A33E7363D}`

Install the provider

1. Install MFA Provider and the providers that are used in the chain, both on the computer with Core Server installed and on the client computer.

   To install MFA Provider, run the file located at `Axidian Access <version number>/Axidian Providers/Axidian MFA Provider` and follow the steps of the installation wizard.

   > ⚠ **IMPORTANT**
   > If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Authentication example

The example uses the Windows Logon component and the Passcode + SMS OTP provider chain.

To sign in with MFA Provider:

1. Select the **Multi-factor authentication** authentication provider.

2. Enter the data for the first provider in the chain.

3. Enter the data for the second provider in the chain and sign in.

> ⓘ **ADDITIONAL SETTINGS**
>
> To configure the authentication parameters, go to the group policies section for MFA.

SMS OTP

The SMS OTP Provider component is designed for user authentication with one-time passwords delivered to users via SMS.

A one-time password is a set of random characters (digits, special characters, and Latin letters). The password is generated on the Axidian Access server, then it is transmitted to the SMS gateway available in the customer infrastructure, after which the one-time code is sent to the user phone number. Data is transmitted through the SMPP (Short Message Peer-to-Peer) protocol.

SMS OTP can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- IIS Extension
- NPS RADIUS Extension
- RDP Windows Logon
- Windows Logon

Provider ID
`{EBB6F3FA-A400-45F4-853A-D517D89AC2A3}`

Prerequisites

To use SMS OTP Provider, an SMS gateway is required. This gateway must be accessible from every Axidian Access server where SMS OTP Provider is to be installed.

To use the provider, the user must have a phone number specified in the default `telephoneNumber` attribute or in another configured attribute. Otherwise, the provider is not available for use.

Authenticator registration in the User Console is not required.

Install the provider

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian SMS OTP Provider/Server/<version number>` and follow the steps of the installation wizard.

> ⚠ **IMPORTANT**
>
> If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.
>
> If the provider is used in client scenarios with Windows Logon and ESSO Agent, install the provider on the client machines with the `Axidian Access <version number>/Axidian Providers/Axidian SMS OTP Provider/Server/<version number>/Client/<Version number>/SMSOTP.Provider.msi` file.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Configure the phone number attribute

To change the default attribute, add the following parameters to the server configuration file `C:\inetpub\wwwroot\am\core\Web.config`:

- The `userMapRules` tag inside the `adUserCatalogProvider` tag
- The `adObjectMapRule` tag inside the `userMapRules` tag with the following parameters:
  - `attribute="Phone"` — specifies the parameter that is changed
  - `adAttribute="mobile"` — specifies the Active Directory attribute that the value is taken from
- The `objectTypeSettings` tag
- The `objectSetting` tag with the `category="person" class="user"` parameters

Example

```
<adUserCatalogProvider id="userId" serverName="axidian.local"
containerPath="DC=axidian,DC=local" userName="userAdmin" password="Q1q2E3e4">
                <userMapRules>
                        <adObjectMapRule attribute="Phone" adAttribute="mobile"/>
                    <objectTypeSettings>
                        <objectSetting category="person" class="user"></objectSetting>
                    </objectTypeSettings>
                </userMapRules>
        </adUserCatalogProvider>
```

Spam protection

The spam protection mechanism is based on calculating the percentage of successful authentications relative to all sent messages over a specified time interval. The calculation process starts only if the number of sent messages exceeds the number that you have specified in the Evaluation Window setting.

When a spam attack is detected, further message sending is blocked for a specified period of time, and a *"Potential spam attack detected"* error occurs when attempting to log in.

Messages can be sent again either after the specified period expires or when a certain percentage of successful authentications is reached.

To configure spam protection, perform the following actions:

1. In Management Console, in the **Configuration→Authenticators** section, select an authenticator.
2. In the **Spam Protection Settings** section, configure the following settings:
    - Enable or disable spam protection.
    - In the **Authentication attempts evaluation window** field, specify the time period during which the percentage of successful login attempts is calculated.
    - In the **Authentication attempts threshold window** field, specify how many login attempts must be made during the time specified in the authentication attempts evaluation window.
    - In the **Percentage of successful authentication attempts** field, specify the minimum percentage of successful logins relative to all sent messages.

Example

The setting is enabled with the following values:

- **Authentication attempts evaluation window** — 600
- **Authentication attempts threshold window** — 20
- **Percentage of successful authentication attempts** — 85

Spam protection is activated if 21 login attempts occur (21 messages sent). The authenticator will be blocked for 600 seconds.

The authenticator will be unblocked in one of the following cases:

- the percentage of successful logins (user successfully entered the one-time password from the message) reaches 85;
- 600 seconds have passed since the blocking.

INFORMATION

Log server events:

- 2090: Potential spam attack detected. Message sending suspended.
- 1118: Message sending to users resumed.

ADDITIONAL SETTINGS

To select and configure the connection type, define the message and phone number format settings, and configure the one-time password settings, go to the group policies section.

Messages Proxy

Messages Proxy is a web application that runs on IIS. Messages Proxy provides a Web API for sending OTPs by running external modules (scripts, binary executable modules). Unlike SMS Proxy, Messages Proxy does not connect to an SMPP gateway; it only runs scripts (any executable file).

System requirements

- Windows Server 2012 R2 and higher
- Internet Information Services 7.5 and higher with the following modules:
  - Web server
  - Static Content Compression
  - HTTP Errors
  - HTTP Redirection
  - Windows Authentication
  - WebDAV Publishing
  - ASP.NET
  - ASP
  - ISAPI Extensions
  - ISAPI Filters
  - IIS Management Console
  - IIS Management Scripts and Tools
  - Management Service

Preconfiguration

1. Install the IIS role on the server where you plan to install Messages Proxy, in one of the following ways:
   - Open **Server Manager**, select **Local Server**, and on the **Manage** tab, click **Add Roles and Features**.
   - Run the `\Misc\IISScripts\Axidian.Main.IIS.Install.MSServer.ps1` script.
2. Install .NET 6 by running `dotnet-hosting-6.0.10-win.exe` from the `Axidian Access <version number>/Axidian Idp` folder of the distribution.

Installation

1. Run the installation file located at `Axidian Access <version number>/Axidian Messages Proxy/<version number>` and follow the steps of the installation wizard.
2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.
3. To remove or restore the product, open the **Control panel** menu and use the standard procedure for the supported operating systems.

Prepare the certificates for Messages Proxy and Core Server

During the Messages Proxy module installation, the SSL certificate requirement is enabled by default, which in turn requires HTTPS configuration.

On the Messages Proxy server (server)

To request a certificate from the Windows domain certification authority:

1. Open the certificate manager **certlm.msc**.
2. In the **Personal** folder, right-click **Certificates** and select **All Tasks** → **Request New Certificate**.
3. In the **Certificate Enrollment** window, click **Next** → **Next**, select the web server certificate template you created earlier, and click **More information is required to enroll for this certificate**.
4. In the **Certificate Properties** window that appears, add the following values and click **OK**:
    - **Subject name**: in the **Type** field, specify *Common name*, and in the **Value** field, specify the full computer name (*Srv01.axidian.local*) or the short one (*Srv01*).
    - **Alternative name**: in the **Type** field, specify *DNS*, and in the **Value** field, specify the full computer name (*Srv01.axidian.local*).
5. Run **IIS Manager**.
6. Go to **Srv01(AXIDIAN\Admin-axidian)** → **Sites** → **Default Web Site** and in the **Actions** section, click **Bindings**.

7. Select the line with https 443 and click **Edit**. In the **Edit Site Binding** window that opens, select your certificate in the **SSL certificate** drop-down list and click **OK**.

8. Select the line with http 80 and click **Remove**.

> ⓘ **INFORMATION**
> Now you can open the IIS start page in a browser at https://Srv01.axidian.local/ (the connection is secure).

On Core Server (client)

Generate a client authentication certificate. You can use the *Computer* template in the Windows domain certification authority as a basis. To do this, on the machine with Core Server:

1. Open the certificate manager **certlm.msc**.

2. In the **Personal** folder, right-click **Certificates** and select **All Tasks** → **Request New Certificate**.

3. In the **Certificate Enrollment** window, click **Next** → **Next** and select the *Computer* certificate template you created earlier.

Edit the configuration file

1. On the Messages Proxy server, open the `app-settings.Production.json` configuration file from the `C:\inetpub\wwwroot\am\proxies\messages` folder.

2. In the `Scripts` field, specify the following values:

   - `Path` — the path to the program that runs the script
   - `Arguments` — the script startup parameters, including the path to the script
   - `Timeout` — the time after which Messages Proxy returns an error if the script has not been executed. The default value is 00:00:20 (20 seconds)

   > ⓘ **INFORMATION**
   > You can add more than one script to the `Scripts` field. To do this, specify several blocks with the parameters for each script, separated by commas. You can also define the script startup order — sequential (`Sequential`) or simultaneous (`Parallel`). If this parameter is not defined, the default value `Sequential` is used.

3. In the `LoggingPermissions` field, specify the parameters that define the information to be logged. The possible parameters are:

   - `Source` — the message sender
   - `Destination` — the message recipient
   - `Error` — the script execution errors
   - `Output` — the messages generated during the script execution
   - `None` (the default value) — the values of all logged fields are hidden
   - `All` — the values of all logged fields are shown

   For example, if the `Source` parameter is specified in `LoggingPermissions`, the *message sender* is logged (not hidden). All the other parameters are hidden and replaced with ****. The logs are located in the `C:\inetpub\wwwroot\am\proxies\messages\Logs\<folder with the date>` folder.

4. The `Publication` field includes the optional `DefaultSource` parameter that contains the default SMS sender name.

5. Optionally, you can configure the parameter that enables Swagger. In the `Documentation` field, set the `Enabled` parameter to `true`. Swagger is then available at `/am/proxies/messages/documentation`.

6. In the `Authentication": { "Certificate": { "Thumbprint": "" } }` field, specify the thumbprint of the (client) certificate that the provider (SMS or Storage SMS) can be authenticated with in Messages Proxy to send messages. You can use several certificates from several clients; in this case the field is named `"Thumbprints"`.

An example of the configuration file

```
{
  "MessagesProxy": {
    "Scripts": [
      {
        "Path": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",
        "Arguments":
            [
                "/c",  "C:\\inetpub\\wwwroot\\am\\proxies\\messages\\script.bat",
                "{0:Source}",
                "{0:Content}",
                "{0:Phone}",
                "{0:RawObjectId}",
                "{0:Sid}",
                "{0:DistinguishedName}",
                "{0:PrincipalName}",
                "{0:SamCompatibleName}",
                "{0:Email}",
                "{0:CustomParameter}"
            ],
        "Timeout": "00:00:10"
      }
    ],
    "LoggingPermissions": "Source"
  },
  "Publication": {
    "DefaultSource": "Axidian sender"
  },
  "Documentation": {
    "Enabled": false
  },
  "Debug": {
    "ExceptionResponsesEnabled": false
  },
  "Authentication": {
    "Certificate": {
      "Thumbprint": "124b3c62f4da3ce8bb271dc4990e354d118daa0g"
    }
  },
  "ReverseProxyIntegration": {
    "BasePath": "/am/proxies/messages"
  },
  "Localization": {
    "DefaultCulture": "en-US"
  }
}
```

Script

The script file reads the parameters passed from Messages Proxy. The parameters are written to the `C:\distr\ps.json` file (the `fileName` parameter from the PowerShell script example). When Messages Proxy is used through an SMS provider, the OTP is passed in the `Content` parameter.

An example of a PowerShell script

```
$body = @(
@{
Source = $args[0]
Content = $args[1]
Phone = $args[2]
RawObjectId = $args[3]
Sid = $args[4]
DistinguishedName = $args[5]
PrincipalName = $args[6]
SamCompatibleName = $args[7]
Email = $args[8]
CustomParameter = "I am PS"
}
)

$body | ConvertTo-Json | Out-File "C:\distr\ps.json"

Invoke-WebRequest -Method Post -Uri 'http://Srv01.axidian.local:82/api/sms' -Body
($body|ConvertTo-Json) -ContentType 'application/json'

Start-Sleep -Seconds 5
```

Configure SMS OTP / Storage SMS OTP Provider

Use group policies to configure SMS OTP Provider or Storage SMS OTP Provider to send OTPs through Messages Proxy. Before you configure the group policies, add the Axidian Access policy templates to the administrative template list. The policy template files are included in the Messages Proxy distribution and are located in the Misc folder.

Select the connection type

With GPO

The *Select the provider connection type* policy applies to the Axidian Access servers and is designed to select the connection type of Core Server (through SMS OTP Provider or Storage SMS OTP Provider) to the SMS gateway.

1. Open the *Select the provider connection type* policy. The policy is located at *Administrative Templates → Axidian-ID → Id Providers → SMS OTP*.
2. Set the policy value to *Enabled*.
3. In the parameters, select the **Messages proxy** option for Messages Proxy.

In the registry

1. Open the registry editor on the Axidian Access server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP` section.
3. Create the *SmsSenderType* parameter of the DWORD type with the value **6**.

Connection type settings

With GPO

The *Messages proxy settings* policy applies to the Axidian Access servers and defines the connection type settings.

1. Open the *Messages proxy settings* policy. The policy is located at *Administrative Templates → Axidian-ID → Id Providers → SMS OTP*.
2. Set the policy value to *Enabled*.
3. Define the following parameters:
   - *URL* — the server connection address
   - *Sender* — the sender name displayed when the message is received

- *Additional text before the OTP* — an arbitrary message text that precedes the one-time password
- *Client certificate thumbprint* — the provider certificate specified in the Messages Proxy `app-settings.Production.json` configuration file (the `Authentication": {"Certificate": { "Thumbprint": "" } }` field)
- *Server certificate thumbprint* — the Messages Proxy certificate for working through HTTPS (configured in IIS)

In the registry

1. Open the registry editor on the Axidian Access server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP\MessagesProxy` section.
3. Create the following parameters:
   - *ServerUrl* — the server connection address
   - *From* (REG_SZ) — the sender name displayed when the message is received
   - *messageOTP* (REG_SZ) — an arbitrary message text that precedes the one-time password
   - *ClientCertificateThumbprint* (REG_SZ) — the provider certificate specified in the Messages Proxy `app-settings.Production.json` configuration file (the `Authentication": { "Certificate": { "Thumbprint": "" } }` field)
   - *ServerCertificateThumbprint* (REG_SZ) — the Messages Proxy certificate for working through HTTPS (configured in IIS)

SMS Proxy

SMS Proxy is a web application that runs on IIS. SMS Proxy connects to the SMPP gateway directly and sends SMS messages. The Axidian Access server does not terminate the session with the SMPP gateway, so the requests are sent within a single connection instead of a separate one for each request.

System requirements

- Windows Server 2012 R2 and higher
- Internet Information Services 7.5 and higher with the following modules:
  - Web server
  - Static Content Compression
  - HTTP Errors
  - HTTP Redirection
  - Windows Authentication
  - WebDAV Publishing
  - ASP.NET
  - ASP
  - ISAPI Extensions
  - ISAPI Filters
  - IIS Management Console
  - IIS Management Scripts and Tools
  - Management Service

Preconfiguration

1. Install the IIS role on the server where you plan to install SMS Proxy, in one of the following ways:
   - Open **Server Manager**, select **Local Server**, and on the **Manage** tab, click **Add Roles and Features**.
   - Run the `\Misc\IISScripts\Axidian.Main.IIS.Install.MSServer.ps1` script.
2. Install .NET 6 by running `dotnet-hosting-6.0.10-win.exe` from the `Axidian Access <version number>/Axidian Idp` folder of the distribution.

Installation

1. Run the installation file located at `Axidian Access <version number>/Axidian SMS Proxy/<version number>` and follow the steps of the installation wizard.
2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.
3. To remove or restore the product, open the **Control panel** menu and use the standard procedure for the supported operating systems.

Prepare the certificate for SMS Proxy

During the SMS Proxy module installation, the SSL certificate requirement is enabled by default, which in turn requires HTTPS configuration.

To request a certificate from the Windows domain certification authority on the SMS Proxy server:

1. Open the certificate manager **certlm.msc**.
2. In the **Personal** folder, right-click **Certificates** and select **All Tasks → Request New Certificate**.
3. In the **Certificate Enrollment** window, click **Next → Next**, select the web server certificate template you created earlier, and click **More information is required to enroll for this certificate**.
4. In the **Certificate Properties** window that appears, add the following values and click **OK**:
   - **Subject name**: in the **Type** field, specify *Common name*, and in the **Value** field, specify the full computer name (*Srv01.axidian.local*) or the short one *Srv01*.
   - **Alternative name**: in the **Type** field, specify *DNS*, and in the **Value** field, specify the full computer name (*Srv01.axidian.local*).
5. Run **IIS Manager**.
6. Go to **Srv01(AXIDIAN\Admin-axidian) → Sites → Default Web Site** and in the **Actions** section, click **Bindings**.
7. Select the line with https 443 and click **Edit**. In the **Edit Site Binding** window that opens, select your certificate in the **SSL certificate** drop-down list and click **OK**.

8. Select the line with http 80 and click **Remove**.

> **INFORMATION**
> Now you can open the IIS start page in a browser at https://Srv01.axidian.local/

Edit the configuration file

1. Open the `app-settings.Production.json` configuration file from the `C:\inetpub\wwwroot\am\proxies\sms` folder.

2. In the `Smpp` section, fill in the following parameters:

   - `SystemId` and `Password` — specify the authentication data to connect to the SMPP server
   - `Host` — specify the connection address of the SMPP server
   - `Port` — specify the port defined for the connection to the SMPP server
   - `SystemType` — an optional sign-in parameter on the SMPP server side, for example, `S1`
   - `EsmeAddressTon` — the possible values are `Unknown`, `International`, `National`, `NetworkSpecific`, `SubscriberNumber`, `Alphanumeric`, `Abbreviated`
   - `EsmeAddressNpi` — the possible values are `Unknown`, `ISDN`, `Data`, `Telex`, `LandMobile`, `National`, `Private`, `ERMES`, `IP`
   - `LoggingPermissions` — specify the parameter that controls whether the sender and recipient information is displayed in the logs. You can specify several values separated by commas. The possible parameters are:
     - `Source` — the message sender
     - `Destination` — the message recipient
     - `None` (the default value) — the values of all logged fields are hidden
     - `All` — the values of all logged fields are shown

3. The `Publication` field includes the optional `DefaultSource` parameter that contains the sender name in the requests, for example, Axidian-AM.

4. In the `Authentication": {"Certificate":}` field, specify the thumbprint of the certificate that the provider (SMS or Storage SMS) can be authenticated with in SMS Proxy to send messages. By default, the configuration for authentication without a certificate is used: `"Certificate": {"Enabled": false, "Thumbprint": "" }`.

> **NOTE**
>
> To rule out possible problems with certificates, first configure the system without them, that is, without establishing a mutual TLS connection:
>
> - When you configure the SMS proxy policy, leave the certificate thumbprint values empty, as in the default settings.
> - In Axidian Access 8.2.6 and higher, additionally add the `"NetworkAddress": { "Enabled": false }` value to the `Authentication": {"Certificate":}` field.

5. To apply the changes in the `app-settings.Production.json` file, restart the *AxidianAM.Core.proxies.sms* server.

An example of the configuration file


```
{
 "Smpp": {
    "SystemId": "smppclient1",
    "Password": "password",
    "Host": "am1.axidian.local",
    "Port": "2776",
    "EsmeAddressTon": "International",
    "EsmeAddressNpi": "ISDN",
    "ReconnectionDelay": "00:02:00",
    "LoggingPermissions": "Source"
    },
 "Publication": {
    "DefaultSource": "Axidian-Id"
    },
 "Documentation": {
    "Enabled": false
    },
 "Debug": {
    "ExceptionResponsesEnabled": false
    },
 "Authentication": {
    "Certificate": {
        "Enabled": false,
        "Thumbprint": "",
        "ForwardingHeader": "Proxy-Client-Certificate"
       },
        "NetworkAddress": {
           "Enabled": false
       }
    },
 "ReverseProxyIntegration": {
    "BasePath": "/am/proxies/sms"
    },
 "Localization": {
    "DefaultCulture": "en-US"
    }
}
```


Select the connection type

INFORMATION

Before you configure the group policy, add the Axidian Access policy templates to the administrative template list. The policy template files are included in the SMS Proxy distribution and are located in the Misc folder.

With GPO

The *Select the provider connection type* policy applies to the Axidian Access servers and is designed to select the Sender of SMS OTP Provider.

1. Open the *Select the provider connection type* policy. The policy is located at *Administrative Templates → Axidian-ID → Id Providers → SMS OTP*.
2. Set the policy value to *Enabled*.
3. In the parameters, select the **SMS proxy** option for SMS Proxy.

In the registry

1. Open the registry editor on the Axidian Access server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP` section.
3. Create the *SmsSenderType* parameter of the DWORD type with the value **5**.

SMS proxy settings

With GPO

The policy applies to the Axidian Access servers and defines the settings of the SMS sending service.

Enable the *SMS proxy settings* policy and specify the required parameters:

- *URL* — the connection address of SMS Proxy. For example: `https://am1.axidian.local/am/proxies/sms`.
- *Sender* — the sender name displayed when the SMS is received.
- *Additional text before the OTP* — an arbitrary message text that precedes the one-time password.
- *Client certificate thumbprint* (an optional parameter) — the provider certificate specified in the SMS Proxy `app-settings.Production.json` configuration file (the `Authentication": {"Certificate": {"Thumbprint": "" } }` field). This certificate is added to the client certificate list when a request is made to SMS Proxy.

- *Server certificate thumbprint* — the SMS Proxy certificate for working through HTTPS (configured in IIS).

In the registry

1. Open the registry editor on the Axidian Access server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP\SMSProxy` section.
3. Create the following parameters:
   - *ServerUrl* — the connection address of SMS Proxy. For example: `https://am1.axidian.local/am/proxies/sms`.
   - *From* (REG_SZ) — the sender name displayed when the message is received.
   - *messageOTP* (REG_SZ) — an arbitrary message text that precedes the one-time password.
   - *ClientCertificateThumbprint* (REG_SZ) — the provider certificate specified in the SMS Proxy `app-settings.Production.json` configuration file (the `Authentication": {"Certificate": {"Thumbprint": "" } }` field).
   - *ServerCertificateThumbprint* (REG_SZ) — the SMS Proxy certificate for working through HTTPS (configured in IIS).

Mutual TLS connection

To establish a mutual TLS connection between Core Server and SMS Proxy:

1. For the SMS Proxy server (server), generate a server authentication certificate (using the *Web Server* template).

2. For Core Server (client), generate a client authentication certificate (for example, using the *Computer* template in the Windows domain certification authority).

   To do this, on the host with Core Server:

   1. Open the certificate manager **certlm.msc**.

   2. In the **Personal** folder, right-click **Certificates** and select **All Tasks** → **Request New Certificate**.

3. In the **Certificate Enrollment** window, click **Next** → **Next** and select the *Computer* certificate template you created earlier.

3. Export the client certificate to a file and install it on the SMS Proxy server in **Computer certificates** → **Personal**.

4. In the SMS Proxy configuration file, specify the following parameters.

```
"Authentication": {
   "Certificate": {
      "Enabled": true,
      "Thumbprint": "client_certificate_thumbprint"
      }
    }
```

In the table below, you can find the possible parameters for the `"Authentication": {"Certificate":}` field.

Parameter	Description
`Authentication:Certificate:Enabled`	`true` by default. Enables the use of a mutual TLS connection.
`Authentication:Certificate:ForwardingEnabled`	`true` by default. Enables support for forwarding the client certificate from the proxy server in the `Authentication:Certificate:ForwardingHeader` HT header.
`Authentication:Certificate:ForwardingHeader`	A Base64-encoded certificate is expected. In an out of-process integration with IIS, the certificate is pass in MS-ASPNETCORE-CLIENTCERT, which is used automatically. Example: Proxy-Client-Certificate
`Authentication:Certificate:Thumbprint`	Example: 380854EA27F2BAED041A941CAC73337E27D0BA
`Authentication:Certificate:RevocationMode`	`NoCheck` by default. The possible values are `NoChec` `Online`, `Offline`.
`Authentication:Certificate:ValidateCertificateUse`	`true` by default. Defines whether the EKU check is required.

Parameter	Description
`Authentication:Certificate:ValidateValidityPeriod`	`true` by default. Defines whether the certificate valic period check is required.

5. Add the certificate thumbprint to the *SMS proxy settings* policy on Core Server.

Storage SMS OTP

If you use Microsoft SQL as the Axidian Access data storage, you can use Storage SMS OTP Provider. This provider allows you to store, retrieve, and update the phone numbers of the Axidian Access users in the database. The phone numbers are stored in encrypted form.

Storage SMS OTP Provider can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- NPS RADIUS Extension
- Windows Logon

Provider ID
`{3F2C1156-B5AF-4643-BFCB-9816012F3F34}`

Prerequisites

To use Storage SMS OTP Provider, an SMS gateway is required. This gateway must be accessible from every Axidian Access server where Storage SMS OTP Provider is to be installed.

Install the provider

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian Storage SMS OTP Provider/Server/<version number>` and follow the steps of the installation wizard.

   > ⓘ **NOTE**
   > If the provider is used in client scenarios with Windows Logon and ESSO Agent, install the provider on the client machines with the `Axidian Access <version number>/Axidian Providers/Axidian Storage SMS OTP Provider/Client/<Version number>/SMSOTP.Provider.msi` file.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Register the provider

We recommend that you register the provider through the lightweight version of the Phone Management Server API, which is designed specifically to work with Storage SMS OTP Provider. You can also use the main Axidian Access API.

Spam protection

The spam protection mechanism is based on calculating the percentage of successful authentications relative to all sent messages over a specified time interval. The calculation process starts only if the number of sent messages exceeds the number that you have specified in the Evaluation Window setting.

When a spam attack is detected, further message sending is blocked for a specified period of time, and a *"Potential spam attack detected"* error occurs when attempting to log in.

Messages can be sent again either after the specified period expires or when a certain percentage of successful authentications is reached.

To configure spam protection, perform the following actions:

1. In Management Console, in the **Configuration→Authenticators** section, select an authenticator.
2. In the **Spam Protection Settings** section, configure the following settings:
   - Enable or disable spam protection.
   - In the **Authentication attempts evaluation window** field, specify the time period during which the percentage of successful login attempts is calculated.
   - In the **Authentication attempts threshold window** field, specify how many login attempts must be made during the time specified in the authentication attempts evaluation window.
   - In the **Percentage of successful authentication attempts** field, specify the minimum percentage of successful logins relative to all sent messages.

Example

The setting is enabled with the following values:

- **Authentication attempts evaluation window** — 600
- **Authentication attempts threshold window** — 20
- **Percentage of successful authentication attempts** — 85

Spam protection is activated if 21 login attempts occur (21 messages sent). The authenticator will be blocked for 600 seconds.

The authenticator will be unblocked in one of the following cases:

- the percentage of successful logins (user successfully entered the one-time password from the message) reaches 85;
- 600 seconds have passed since the blocking.

INFORMATION

Log server events:

- 2090: Potential spam attack detected. Message sending suspended.
- 1118: Message sending to users resumed.

ADDITIONAL SETTINGS

To configure the SMS delivery service, as well as the message and one-time password format, go to the group policies section.

OMNIKEY

The component enables HID OMNIKEY smart card readers to work with the Axidian Access modules. For more information about OMNIKEY readers and cards, visit the manufacturer website.

> **NOTE**
> Authenticators of this type can only be registered with the authenticator management tool.

OMNIKEY Provider is supported in the following modules:

- Enterprise SSO
- Windows Logon

The following HID OMNIKEY reader models are currently supported:

- OMNIKEY 5325
- OMNIKEY 6321
- OMNIKEY 5321 CLi
- OMNIKEY 6321 CLi
- OMNIKEY 5427 CK
- OMNIKEY 5421
- OMNIKEY 5022
- OMNIKEY 5025

Provider ID
`OMNIKEY {4B15AF52-A795-4CA6-B7CD-CDB8ABF2D2C2}` `OMNIKEY + PIN {199B8FA8-FA9D-4ED8-941D-F86A5C681E0C}`

Prerequisites

The provider requires Bsp Broker installed on the Axidian Access server.

Files for Bsp Broker are located at `Axidian Access <version number>\Axidian Providers\Axidian Bsp Broker\<Version number>`.

- *AuthProviders.BspBroker-<version number>.x64.en-us.msi* — the installation package of Bsp Broker for 64-bit operating systems.
- *AuthProviders.BspBroker-<version number>.x86.en-us.msi* — the installation package of Bsp Broker for 32-bit operating systems.

Install the provider

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian OMNIKEY Provider/<version number>` and follow the steps of the installation wizard. Install the provider both on the computer with Core Server installed and on the client computer.

   > ⚠ **IMPORTANT**
   > If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Authentication example

1. When you first sign in to the system or to an application via Axidian Access authentication, select the sign-in method.

   To do this, click **Change the sign-in method** in the **Windows sign-in** window (in the **Authentication for Enterprise SSO** window). Select the *Card (HID OMNIKEY)* sign-in method.

2. Connect the HID OMNIKEY card reader and place a registered card on the reader.

3. The authentication is performed once the card data has been processed. If the sign-in with the authenticator is successful, the *Card* sign-in method is saved as the preferred one and is offered to the user automatically at the next sign-in to the system or to an application.

ADDITIONAL SETTINGS

To define the smart card removal behavior, configure the timeout of the action performed on smart card removal, and allow or forbid the use of PACS data, go to the group policies section for OMNIKEY Provider.

Futronic

Futronic Provider is required for user authentication with the Futronic FS80 biometric scanner. For more information about Futronic devices, visit the official Futronic Technologies website.

> **NOTE**
> Authenticators of this type can only be registered with the authenticator management tool.

Futronic Provider is supported in the following modules:

- Enterprise SSO
- Windows Logon

Provider ID
`{A0EF00AD-1EEB-4D48-8BCF-06E19CD5585F}`

Prerequisites

The provider requires Bsp Broker installed on the Axidian Access server.

Files for Bsp Broker are located at `Axidian Access <version number>\Axidian Providers\Axidian Bsp Broker\<Version number>`.

- *AuthProviders.BspBroker-<version number>.x64.en-us.msi* — the installation package of Bsp Broker for 64-bit operating systems.
- *AuthProviders.BspBroker-<version number>.x86.en-us.msi* — the installation package of Bsp Broker for 32-bit operating systems.

Install the provider

1. Run the installation file located at `Axidian Access <version number>/Axidian Providers/Axidian Futronic Provider/<version number>` and follow the steps of the installation wizard. Install the provider both on the computer with Core Server installed and on the client computer.
2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Authentication example

1. When you first sign in to the system or to an application via Axidian Access authentication, select the sign-in method.

   To do this, click **Change the sign-in method** in the **Windows sign-in** window. Select the *Finger* authentication method.

2. Connect the biometric scanner to the computer and place your finger on the scanner.

3. If the sign-in with the authenticator is successful, the *Finger* sign-in method is saved as the preferred one and is offered to the user automatically at the next sign-in to the system or to an application.

> ⓘ **ADDITIONAL SETTINGS**
>
> To configure the minimum acceptable template quality when enrolling, the maximum acceptable FAR, and the scanner indication parameters, go to the group policies section for Futronic Provider.

Telegram

Telegram Provider is designed for user authentication with the following technologies:

- One-time passwords
- Push notifications with a request to confirm or reject the sign-in

Users receive the one-time passwords and the push notifications through a bot in the Telegram messenger.

Telegram Provider can be used for authentication in the following modules:

- ADFS Extension
- Identity Provider
- IIS Extension (in the one-time password mode only)
- NPS RADIUS Extension (in the one-time password mode only)
- Windows Logon (in the one-time password mode only)

> ⓘ **NOTE**
> The component requires access to the Telegram servers both for the service and for the provider.

Provider ID
{CA4645CC-5896-485E-A6CA-011FCC20DF1D}

Preconfiguration

To configure Telegram Provider:

1. Install the provider.
2. Install the service.
3. Configure the phone number attribute.
4. Create a bot in the Telegram application.
5. Configure the service.
6. Define the settings in the Management Console.

7. Register the authenticator.

Install the provider

1. Run the installation file located at `axidian <Version number>/Axidian Providers/Axidian Telegram OTP Provider/<Version number>`.

> ⚠ **IMPORTANT**
> If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.

2. After the component installation is complete, you must restart the system. In the installation wizard window, click **Yes** to restart immediately, or **No** to do it later manually.

Install the service

1. Run the `Telegram.Service-<version number>.x86.en-us.msi` file located at `axidian <Version number>/Axidian Providers/Axidian Telegram OTP Provider/`.

2. After the component installation is complete, you must restart the system. In the installation wizard window, click **Yes** to restart immediately, or **No** to do it later manually.

Configure the phone number attribute

To use the provider, specify the phone number in an Active Directory user attribute. By default, this is the `telephoneNumber` attribute. You can also configure any other attribute. If the number is not specified, the provider cannot be used.

> ⚠ **IMPORTANT**
> Before you configure the attribute, be sure to decrypt and back up the Core Server configuration file `inetpub/wwwroot/am/core/Web.config`.

To configure the phone number attribute:

With the configuration wizard

1. Run the configuration wizard on the computer with Core Server installed: *Start* → *Axidian* → *Axidian Access configuration wizard*.
2. Go to the **User catalogs** tab.
3. Select a catalog.
4. Click **Edit**.
5. A warning appears stating that if you reduce the catalog scope or change the domain, the user data becomes unavailable. If you have already decrypted and backed up the `inetpub/wwwroot/am/core/Web.config` file, click **OK**.
6. In the configuration wizard, click **Configure attribute mapping**.
7. In the **Phone** field, enter the name of the Active Directory user attribute that contains the phone number, for example, the *mobile* attribute.
8. Click **Save** twice.
9. Go to the **Confirmation** tab and click **Apply**.
10. After the configuration file is set up and the operation is verified, close the wizard.

With the configuration file

1. Open the Core Server configuration file `inetpub/wwwroot/am/core/Web.config`.
2. Add the following to the `adUserCatalogProvider` section:

```
            <userMapRules>
                    <adObjectMapRule attribute="Phone"
adAttribute="attribute_name_in_Active_Directory"/>
                <objectTypeSettings>
                    <objectSetting category="person" class="user"></objectSetting>
                </objectTypeSettings>
            </userMapRules>
```

   where:

   - `attribute="Phone"` — the attribute name in Axidian Access
   - `adAttribute="attribute_name_in_Active_Directory"` — the name of the corresponding Active Directory attribute that the phone number value is taken from

Example

```
<adUserCatalogProvider id="userId" serverName="axidian.local"
containerPath="DC=axidian,DC=local" userName="userAdmin" password="Q1q2E3e4">
                //highlight-grey-start
                <userMapRules>
                        <adObjectMapRule attribute="Phone" adAttribute="mobile"/>
                    <objectTypeSettings>
                        <objectSetting category="person" class="user"></objectSetting>
                    </objectTypeSettings>
                </userMapRules>
                //highlight-grey-end
        </adUserCatalogProvider>
```

Create a bot

1. Open the Telegram application, find the dedicated @botfather bot, and open a dialog with it.
2. Click **Start** and send the `/newbot` command.
3. Specify the name of the bot you create. This name is displayed in the dialog window with the bot. Enter any name, for example, *AxidianOTP*.
4. Specify the username for the bot account, ending with *bot*, for example, *AxidianOTP_bot*. This username is used for links to the bot.
5. If the registration is successful, the API access token is displayed. You need this token to configure the *AM Telegram Service* service and the bot in the Management Console.

Configure the service

To configure the service, you need a service user with the global administrator rights. Authenticators are registered on behalf of this user. For security reasons, we recommend that you create a separate user in Active Directory for the Telegram Provider registration.

To configure the Telegram Service:

1. Open the `AM.Telegram.Service.exe.config` service configuration file located at `C:\Program Files (x86)\Axidian-Id\Axidian-Id Telegram Service`.

   To save the changes in the configuration file, run the editor with administrator rights.

2. In the `appSettings` section, set the values in the `value` parameters for the corresponding `key` parameters:

`culture`	The localization language; Russian and English are supported. The default value is *ru-RU*.
`EAWebAPIURL`	The Core Server URL.
`username`	The name of the service user with the global administrator rights on whose behalf the authenticators are registered.
`password`	The service user password.
`lognames`	The user name logging format. The default value is *Name*. **Supported formats** - `Id` — the user ID in Axidian Access in the *rootUserCatalogProviderId_Guid* format - `Name` — the value of the `name` attribute from Active Directory - `CanonicalName` — the user name in the *Canonical-Name* format (myserver.demo.local/users/UserName) - `PrincipalName` — the value of the `userPrincipalName` attribute from Active Directory - `SamCompatibleName` — the user name in the *domainName\UserName* format - `DistinguishedName` — the value of the `distinguishedName` attribute from Active Directory - `Sid` — the value of the `objectSid` attribute from Active Directory - `Email` — the value of the `mail` attribute from Active Directory, or the value specified in the Core Server configuration file - `Phone` — the value of the `telephoneNumber` attribute from Active Directory, or the value specified in the server configuration file **Example:** `add key="lognames" value="PrincipalName,Sid"`
`botId`	The token received when you created the bot in Telegram.
`trustedId`	An arbitrary unique identifier. The identifier value must match the one specified in the *Telegram Service trusted ID* setting in the Management Console.

An example of the edited `appSettings` section

```
      <appSettings>
          <add key="culture" value="en-US" />
          <add key="EAWebAPIURL" value="https://server.axidian.local/am/core/" />
          <add key="username" value="telegram@axidian.local" />
          <add key="password" value="Q1q2E3e4" />
          <add key="lognames" value="Name" />
          <add key="botId" value="1156320278:AAG24_EODMotm_feYYQOfbPddK_2Z_JDiKQ" />
          <add key="trustedId" value="the unique identifier set when configuring the
Telegram Service" />
        </appSettings>
```

3. In the `logServer` section, specify the values of the following parameters:

   - In the `URL` parameter, specify the URL to connect to Log Server in the `http(s)://full_dns_server_name/ls/api` format.
   - If the private key is in the registry and the certificate is in the computer storage, specify the certificate thumbprint in the `CertificateThumbprint` parameter.
   - If the key pair is in a pfx file, specify the path in the `CertificateFilePath` parameter and the pfx password in the `CertificateFilePassword` parameter.

     Example

     ```
     <logServer Url="https://logserver.axidian.local/ls/api/"
     CertificateThumbprint="YOUR_CERTIFICATE_THUMBPRINT" CertificateFilePath="PATH_TO_PFX"
     CertificateFilePassword="PFX_PASSWORD" />
     ```

4. Save the changes.

5. Start the AM Telegram Service.

How to start the service

1. Run Task Manager.
2. In the window that opens, click **More details** and go to the **Services** tab.
3. Find **AM Telegram Service** in the list, right-click it, and select **Start**.

> **NOTE**
>
> If you use several Core Servers, the service can only run on one of them at a time.

Define the settings in the Management Console

For the provider to work correctly, define the remaining settings in the Management Console. To do this:

1. In the Management Console side panel, open the **Configuration** section.
2. Go to the **Authenticators** tab.
3. Select the Telegram authenticator.
4. If required, define the general settings.
5. In the **Telegram Service settings** section, specify an arbitrary unique identifier. The identifier value must match the one specified in the `AM.Telegram.Service.exe.config` service configuration file.
6. In the **Operating mode** section, select the provider operating mode — *Push* or *OTP*.
7. In the **Bot settings** section:
   - In the **Bot ID** field, specify the token received when you created the bot in Telegram.
   - In the **Message template** field, configure the message appearance. The message can contain the one-time code to sign in to the application, the application name, the user name, the sign-in date and time, and the IP address of the computer used for the sign-in.
8. If you plan to use Telegram Provider in the one-time password mode, in the **One-time password generation settings** section:
   - In the **One-time password length** field, specify how many characters the one-time password contains.
   - In the **Digits** setting, specify whether the one-time password contains digits.
   - In the **Lowercase Latin letters** setting, specify whether the one-time password contains lowercase Latin letters.
   - In the **Uppercase Latin letters** setting, specify whether the one-time password contains uppercase Latin letters.

- In the **Special characters** setting, specify whether the one-time password contains special characters.

9. If required, configure the following:

Configure Telegram Provider to work through a proxy

In the **Bot settings** section:

- In the **Use proxy** setting, select *Yes*. *No* is selected by default.
- In the **Proxy address** field, specify the proxy server address.

Configure spam protection

The spam protection mechanism is based on calculating the percentage of successful authentications relative to all sent messages over a specified time interval. The calculation process starts only if the number of sent messages exceeds the number that you have specified in the Evaluation Window setting.

When a spam attack is detected, further message sending is blocked for a specified period of time, and a *"Potential spam attack detected"* error occurs when attempting to log in.

Messages can be sent again either after the specified period expires or when a certain percentage of successful authentications is reached.

To configure spam protection, perform the following actions:

1. In Management Console, in the **Configuration→Authenticators** section, select an authenticator.
2. In the **Spam Protection Settings** section, configure the following settings:
   - Enable or disable spam protection.
   - In the **Authentication attempts evaluation window** field, specify the time period during which the percentage of successful login attempts is calculated.
   - In the **Authentication attempts threshold window** field, specify how many login attempts must be made during the time specified in the authentication attempts evaluation window.
   - In the **Percentage of successful authentication attempts** field, specify the minimum percentage of successful logins relative to all sent messages.

Example

The setting is enabled with the following values:

- **Authentication attempts evaluation window** — 600
- **Authentication attempts threshold window** — 20
- **Percentage of successful authentication attempts** — 85

Spam protection is activated if 21 login attempts occur (21 messages sent). The authenticator will be blocked for 600 seconds.

The authenticator will be unblocked in one of the following cases:

- the percentage of successful logins (user successfully entered the one-time password from the message) reaches 85;
- 600 seconds have passed since the blocking.

INFORMATION

Log server events:

- 2090: Potential spam attack detected. Message sending suspended.
- 1118: Message sending to users resumed.

Register the authenticator

NOTE

During the provider registration, the user is asked for the phone number from their Telegram account. The user phone number in Active Directory must match it exactly and be in the same format. The supported formats are +7xxxxxxxxxx and 8xxxxxxxxxx.

1. Open the chat with the bot you created. To do this, enter the name of your bot in the messenger search bar, or follow the link from the BotFather message.
2. Click **Start**.
3. Enter the `/register` command.

4. Enter the phone number of the Telegram account.

5. If the authenticator registration is successful, the *You have been successfully registered* message is displayed.

Secured TOTP

Secured TOTP is a provider bound to the user device ID.

The secret key used to generate the one-time code is encrypted and can only be decrypted with a key based on the device ID. Therefore, a particular authenticator can only be registered on a single device.

Secured TOTP can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- NPS RADIUS Extension
- RDP Windows Logon
- Windows Logon

Prerequisites

The provider requires the following conditions:

- Secured TOTP Provider is installed.
- The Axidian Key mobile application 2.6 or higher is installed.
- To register Secured TOTP via a link from an email, the following is required:
  - Email OTP Provider is installed and configured.
  - The user email is filled in the Active Directory catalog.

Provider IDs for integration with the NPS Radius Extension module

```
{F15FD7EC-19EA-4384-846E-A2D0BE149FA2} – Secured TOTP
{882C1787-FD32-44A2-BA89-F1F529FBE7AB} – Passcode + Secured TOTP
{7F3DE86F-59D1-4476-AA5D-F277E5DD5938} – Windows Password + Secured TOTP
```

Install the provider

To install Secured TOTP, run the `AuthProviders.SecuredTOTP-<version number>.<bitness>.en-us.msi` package from the `axidian <Version number>\Axidian Providers\Axidian Secured TOTP Provider\<Version`

`number>` folder.

> ⚠ **IMPORTANT**
> If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.

After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Access rights

To grant or revoke the rights to use the authenticator:

1. In the Management Console, go to **Configuration** → **Authenticators** and select the Secured TOTP authenticator from the list.
2. In the **General settings** section, configure the following:
   - In the **Forbid to use** setting, specify whether the user can use or register the selected authenticator.
3. In the **Actions available to the user** section, configure the following:
   - In the **Register new authenticators** setting, specify whether the user can register new authenticators.
   - In the **Edit existing authenticators** setting, specify whether the user can modify the authenticators that are already registered.
   - In the **Delete existing authenticators** setting, specify whether the user can delete the authenticators that are already registered.
   - In the **Allow editing the authenticator comment** setting, specify when the user can edit the comments of the authenticators that are already registered.

Block the authenticator

To block the authenticator:

1. In the Management Console, go to **Configuration** → **Authenticators** and select the Secured TOTP authenticator from the list.

2. In the **Authenticator blocking settings** section, configure the following:
    - Allow or block the use of Secured TOTP in case of a series of failed authentication attempts.
    - In the **Number of authentication attempts before blocking** field, specify how many times the user can fail authentication before the authentication method is blocked.
    - In the **Blocking counter reset** field, specify how many minutes must pass after a failed authentication attempt before the counter is reset.
    - In the **Timeout before the sign-in method is unblocked** field, specify after how many minutes the blocked authentication method becomes available again.

One-time password settings

The settings are applied to the Axidian Access servers and allow you to define the length of the one-time password and the character groups it contains.

To configure the one-time password generation:

1. In the Management Console, go to **Configuration → Authenticators** and select the Secured TOTP authenticator from the list.
2. In the **One-time password generation settings** section, configure the following:
    - In the **One-time password length** field, specify how many characters the one-time password contains.
    - In the **Digits** setting, specify whether the one-time password contains digits.
    - In the **Lowercase Latin letters** setting, specify whether the one-time password contains lowercase Latin letters.
    - In the **Uppercase Latin letters** setting, specify whether the one-time password contains uppercase Latin letters.
    - In the **Special characters** setting, specify whether the one-time password contains special characters.

Registration in the Management Console

Secured TOTP can be registered via a link from an email or with a QR code. Both registration methods are available in the Management Console.

Registration via a link from an email requires the following conditions:

- Email OTP Provider is installed and configured.
- The Axidian Key mobile application 2.6 or higher is installed.
- The user email is filled in the Active Directory catalog.

The link is sent to the email address specified for the user in the Active Directory catalog. Optionally, you can duplicate the link to another email address, for example, to the employee manager.

To register the authenticator by email:

1. In the Management Console, in the **Authenticator registration method in the Management Console** setting, select the registration by email.
2. Specify the email subject text.
3. Specify the email message text. The registration link is specified with the special <*regLink*> tag.
4. Specify the link text to replace the explicit spelling.
5. Then go to the **Users** section, select a user, and open the **Authenticators** tab.
6. Click **Register** and select *Secured TOTP* from the list.
7. In the window that opens, enter the user device ID. You can obtain it in the settings of the Axidian Key application and copy it or send it with the **Share** button. Optionally, at this step you can specify additional email addresses, separated by commas.
8. Click **Next**. After that, the user receives an email with a link to the authenticator registration. Following the link, the user goes to the Axidian Key application, where the authenticator registration starts.
9. In the Management Console, click **Save**.

Configuring the Secured TOTP registration link (and the Axidian Key download link)

The link to the Secured TOTP authenticator registration and to the Axidian Key application download can be defined manually.

1. Go to **Configuration** → **Authenticators**, to the **Registration settings** section.

2. In the **Server address used to form the authenticator registration link (deeplink)** setting, specify the registration link that is sent by email.

   The following registration links are available (the server location is given in brackets):

   - https://indeedkey.drru.agconnect.link (Russia)
   - https://indeedkey.dre.agconnect.link (Germany)

- https://indeedkey.drcn.agconnect.link (China)
- https://indeedkey.dra.agconnect.link (Singapore)

3. In the **Additional settings** section, specify the link to download Axidian Key.

   The following download links are available (the server location is given in brackets):

   - https://indeedkey.drru.agconnect.link/XwZr (Russia)
   - https://indeedkey.dre.agconnect.link/MJyW (Germany)
   - https://indeedkey.drcn.agconnect.link/NDbK (China)
   - https://indeedkey.dra.agconnect.link/i1RD (Singapore)

Forced authenticator verification

You can configure forced authenticator verification for the authenticator registration.

1. Go to **Configuration** → **Authenticators** and select **Secured TOTP** from the list.
2. In the **General settings** section, enable the forced verification. An additional confirmation window for entering the authentication data appears during registration.
3. Enter the authentication data you have received and click **Confirm**.

Registration in the User Console

A user can register a new authenticator in the User Console if the administrator has allowed the user to register new authenticators in the Secured TOTP settings in the Management Console.

The Axidian Key application 2.6 or higher must be installed on the user device.

To register the authenticator with a QR code:

1. In the Management Console, in the **Actions available to the user** section, allow the user to register new Secured TOTP authenticators.
2. In the **Authenticator registration method in the User Console** setting, select the registration with a QR code.
3. Secured TOTP appears in the list of available authenticators for the user in the User Console.

The user must do the following:

1. Click the gear icon.

2. Click **Register**.
3. Specify the device ID. To obtain the ID, the user must open the Axidian Key application, click the gear icon, and copy the value from the **Device ID** field.
4. Scan the code that appears with the device where the Axidian Key application is open.
5. If forced authenticator verification is configured, additionally enter the one-time password received.

IronLogic Z2 USB

IronLogic Z2 USB Provider is required for user authentication with IronLogic readers.

> ⚠ **IMPORTANT**
> The provider requires a dedicated version of the IronLogic Z2 USB reader. Other versions of Z2 USB are not guaranteed to work with Axidian Access.

For more information about IronLogic readers and tags, visit the manufacturer website.

> ⓘ **NOTE**
> Authenticators of this type can only be registered with the authenticator management tool.

IronLogic Z2 USB Provider is supported in the following modules:

- Enterprise SSO
- Windows Logon

Provider ID
`Z2 USB {CB5109DA-B575-422C-8805-524FE12B02F5}` `Z2 USB + PIN {1EDFD6E1-FD66-4A5B-971A-CBA0C611BA9B}`

Prerequisites

The provider requires Bsp Broker installed on the Axidian Access server.

Files for the Bsp Broker installation are located at `axidian\Axidian Providers\Axidian Bsp Broker\<Version number>\`.

- `AuthProviders.BspBroker.x64.msi` — the installation package of Bsp Broker for 64-bit operating systems.
- `AuthProviders.BspBroker.x86.msi` — the installation package of Bsp Broker for 32-bit operating systems.

Install the provider

1. Run the installation file located at `axidian <Version number>/Axidian Providers/Axidian IronLogic Z2USB Provider/<Version number>` and follow the steps of the installation wizard. Install the provider both on the computer with Core Server installed and on the client computer.

   > ⚠ **IMPORTANT**
   > If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Authentication example

1. When you first sign in to the system or to an application via Axidian Access authentication, select the sign-in method.

   To do this, click **Change the sign-in method** in the **Windows sign-in** window. Select the *Card (Z2 USB Card Reader)* authentication method.

2. Connect the Z2 USB card reader and place a registered card on the reader.

3. The authentication is performed once the card data has been processed. If the sign-in with the authenticator is successful, the *Card* sign-in method is saved as the preferred one and is offered to the user automatically at the next sign-in to the system or to an application.

> ⓘ **ADDITIONAL SETTINGS**
> To define the smart card removal behavior, configure the timeout of the action performed on smart card removal, and specify the storage format of card IDs in the Axidian Access database, go to the group policies section for IronLogic Z2 USB Provider.

Hardware TOTP

Hardware TOTP can be used for authentication in the following modules:

- ADFS Extension
- Enterprise SSO
- Identity Provider
- IIS Extension
- NPS RADIUS Extension
- RDP Windows Logon
- Windows Logon

Hardware TOTP allows you to obtain one-time passwords with the following devices:

- The eToken PASS standalone one-time password generator. It can be used for authentication in any applications and services, including those that support the RADIUS authentication protocol — VPN, Microsoft ISA, Microsoft IIS, Outlook Web Access, and others.

Device characteristics

Device	File format	Algorithm	Refresh period
eToken PASS	XML	SHA1	30 seconds

Provider ID
`{CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}`

Install the provider

1. Run the installation file located at `axidian <Version number>/Axidian Providers/Axidian Hardware TOTP Provider/<Version number>` and follow the steps of the installation wizard.
2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.
3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Add a device

You can add a device manually or by uploading a file with the device parameters.

A device can be registered for one user only.

Add manually

To add an eToken PASS device:

1. Open the Management Console.
2. In the menu on the left, select **Devices**.
3. Click **Add device**.
4. In the window that opens, select the eToken PASS model.
5. Fill in the **Serial number**, **Secret key**, and **Comment** fields (the last one is optional) and click **Add**.

Add from a file

To add an eToken PASS device:

1. Open the Management Console.
2. In the menu on the left, select **Devices**.
3. Click **Add from file**.
4. In the window that opens, select the eToken PASS model.
5. In the window that opens, select the XML file with the device parameters and click **Add**.

Synchronize a device

To synchronize a device:

1. Open the Management Console.
2. In the menu on the left, select **Devices**.
3. Select the device from the list and click **Synchronize**.
4. In the new window that opens, in the **One-time password 1** and **One-time password 2** fields, enter the passwords from the device and click **Synchronize**.

Change the serial number and disable a device

To edit a device:

1. Open the Management Console.
2. In the menu on the left, select **Devices**.
3. In the **Authentication provider** drop-down list, select *Hardware TOTP*, in the **Serial number** field enter the device serial number if you know it, and click **Search**.
4. Select the device found and click the edit icon.

Example

5. In the editing window, you can change the device serial number or the comment, or disable the device. After you make the changes, click **Save**.

Example

Delete a device

You can delete a device manually or by uploading a file with the device parameters.

Delete manually

To delete a device:

1. Open the Management Console.
2. In the menu on the left, select **Devices**.
3. In the **Authentication provider** drop-down list, select *Hardware TOTP*, in the **Serial number** field enter the device serial number if you know it, and click **Search**.
4. Select the device found and click **Delete device**.
5. In the window that opens, confirm the deletion.

Delete from a file

To delete an eToken PASS device:

1. Open the Management Console.
2. In the menu on the left, select **Devices**.
3. Click **Delete from file**.
4. In the window that opens, select the eToken PASS model.
5. In the window that opens, select the XML file with the device parameters and click **Delete**.

Smart Card

The Smart Card Provider component is designed for user authentication with personal devices (smart cards, USB keys).

> **NOTE**
> Authenticators of this type can only be registered with the authenticator management tool.

This authentication method is supported in the following modules:

- Enterprise SSO
- Windows Logon

The following devices are supported:

eToken devices

- eToken Pro 32k
- eToken Pro 64k
- eToken Pro Java 72k

ESMART devices

- Token 64k
- Token SC 64k
- Token USB 64k

Avest devices

- Avest Key 256A

SafeNet devices

- iKey 1000
- iKey 1032

JaCarta devices

- JaCarta PKI

Gemalto devices

- IDPrime MD

The same authentication device can be used either with or without a PIN code request:

- The Smart Card Provider component allows the user to authenticate with a card without a PIN code request.
- The Smart Card Provider + PIN component requires a PIN code at every user authentication.

> **NOTE**
> Smart Card Provider with PIN does not allow you to set, change, or delete the PIN codes of the authentication devices. To change or delete the PIN codes, use the dedicated software from the authentication device manufacturers.

Provider ID
`Smartcard {0AF65AD8-DB77-4B64-B489-958D9B36E28C}` `Smartcard + PIN {42456525-8EC1-4AB1-97B8-45AF8635D10F}`

Prerequisites

The provider requires Bsp Broker installed on the Axidian Access server.

Files for Bsp Broker are located at `axidian <Version number>/Axidian Providers/Axidian Bsp Broker/<Version number>`.

- *AuthProviders.BspBroker-<version number>.x64.en-us.msi* — the installation package of Bsp Broker for 64-bit operating systems.
- *AuthProviders.BspBroker-<version number>.x86.en-us.msi* — the installation package of Bsp Broker for 32-bit operating systems.

Install the provider

1. Run the installation file located at `axidian <Version number>/Axidian Providers/Axidian SmartCard Provider/<Version number>` and follow the steps of the installation wizard. Install the provider both on the computer with Core Server installed and on the client computer.

> ⚠ **IMPORTANT**
> If several Core Servers are used in your infrastructure, install the provider on all the servers of the infrastructure.

2. After the installation is complete, a system restart may be required. If the installation wizard prompts you to restart the system, confirm this action.

3. To remove or restore the product, open the Control panel menu and use the standard procedure for the supported operating systems.

Authentication example

1. When you first sign in to the system or to an application via Axidian Access authentication, select the sign-in method.

   To do this, click **Change the sign-in method** in the **Windows sign-in** window (in the **Authentication for Enterprise SSO** window). Select the *Smart card or USB key* or *Smart card or USB key + PIN* sign-in method.

2. Connect the USB key or the smart card to the computer. If the Smart Card Provider + PIN sign-in method is used, enter the PIN code and click **Sign in**.

3. The authentication is performed once the card data has been processed. If the sign-in with the authenticator is successful, the *Smart card or USB key (Smart card or USB key + PIN)* sign-in

method is saved as the preferred one and is offered to the user automatically at the next sign-in to the system or to an application.

Migrate provider settings

In Axidian Access 8.2 and higher, the authentication provider settings are located in the Management Console. When you update from earlier versions, you can transfer the provider settings that were previously defined in policies and in the registry with the *EA.Settings.Migration.Tool* utility. This utility also allows you to display the forced authenticator verification setting.

When settings are migrated from policies, the authenticator settings of the policy with the highest priority are transferred.

You can transfer the settings of the following providers:

- Email OTP
- Axidian Key
- SMS OTP
- Storage SMS OTP
- Telegram

The following settings are transferred:

- The authenticator actions available to the user that require registration.
- The maximum number of authenticators that require registration, as defined in the policy with the highest priority.
- The settings of the **Forbid to use** option for all the installed authenticators, as defined in the policy with the highest priority.

Preparation

Before you migrate the settings:

1. Decrypt the Core Server configuration file `C:\inetpub\wwwroot\am\core\Web.config` in one of the following ways:
   - With the configuration wizard. Open the `C:\inetpub\wwwroot\am\core\Web.config` file, go through all the steps again, and at the **File encryption** step, clear the **Encrypt the configuration file (recommended)** option.
   - With the `decryptConfigs.bat` script. For more information, see Core Server.

2. Create a backup copy of the database.
3. Create backup copies of the configuration files and of the registry of the server components, either manually or with the UpdateHelper utility.
4. If you migrate the settings from the registry, make sure that a group policy with the registry settings is applied on the server with Core Server installed.
5. If you migrate the settings from a policy, create a backup copy of the `C:/inetpub/wwwroot/am/core` folder for Axidian Access 8.1.x before you update the server to 8.2.x. This folder is used as the data source.

Migration

To transfer the settings with the utility:

1. Run the `axidian 8.2.x\Axidian Access Manager Server\8.2.x\Misc\EA.Settings.Migration.Tool\EA.Settings.Migration.Tool.exe` file.
2. Confirm that you have created a backup copy of the database.
3. Enter the Core Server address in the `http(s)://full_dns_server_name/am/core/` format.
4. If required, enter the credentials for access to Core Server.
5. Specify the path to the Core Server configuration file.
6. Use the arrow keys to select one of the options and perform the actions required for it:
    - **Auth method settings** — migration of the settings from the registry. After you select this option, select the providers whose settings you want to transfer, using the spacebar or the *Ctrl+A* key combination (to select all the providers).
    - **Policy settings** — migration of the settings from a policy. After you select this option, enter the path to the backup copy of the Core Server `Web.config` configuration file. Then select the policy whose settings you want to transfer.
    - **Forced Authenticator Verification policy** — to display the forced authenticator verification setting in the Management Console.

> **NOTE**
> The forced authenticator verification setting is displayed only for the one-time password authentication providers that have already been installed. After you install new providers, run the migration again.

7. Press **Enter**.

8. In the log that is displayed, make sure that the settings have been migrated successfully.

> **IMPORTANT**
> After you transfer the settings, check the number of authenticators in the **Maximum number** setting for each provider in the Management Console.

Group policy administrative templates (ADMX)

Some Axidian Access settings are specified with Microsoft group policies.

> **NOTE**
> Before configuring group policies, add the Axidian Access policy templates to the list of administrative templates. The policy template files are included in the Axidian Access distribution and are located in the `Misc\ADMX Templates` directory.

The effect of a policy is adding certain keys to the registry; if required, the policy values can be added to the registry manually. When adding them manually, create the missing registry sections.

Add administrative templates

1. Copy the group policy template files from the `C:\Axidian\Axidian Access <version number>\Misc\ADMX Templates` distribution directory to one of the following directories:

   - `C:\Windows\PolicyDefinitions` — the local storage of ADMX files.
   - `C:\Windows\SYSVOL\sysvol\DOMAIN_NAME\Policies\PolicyDefinitions` — the storage of ADMX files on the domain controller.

   If the *PolicyDefinitions* directory does not exist, create it.

2. Copy the `AxidianID.BaseFile.admx` file and all the required ADMX files to the `\PolicyDefinitions` directory.

3. Copy the `AxidianID.BaseFile.adml` file and all the ADML files corresponding to the copied ADMX files from the `\Axidian Access <version number>\Misc\ADMX Templates\<en-US or ru-RU>` directory to the `\PolicyDefinitions\en-US` directory (for the English server localization) or `\PolicyDefinitions\ru-RU` (for the Russian server localization).

4. Copy the templates from the `ADMX Templates\BSPs` folder to the root *PolicyDefinitions* folder, next to the other ADMX files.

5. Add the policy templates and the authentication provider templates to the Computer Configuration section. Add the Axidian Access Paste policy templates to the User Configuration section.

After they are loaded, the Axidian Access group policies become available in Group Policy Management Editor, in the ADMX files section.

Configure with Local Group Policy Editor

When configuring policies, you can use the gpedit.msc group policy editor console to manage the parameters in the registry.

Configure with Registry Editor

To configure policies manually, open Registry Editor:

1. In the search box on the taskbar, enter regedit, then select Registry Editor.
2. Go to the required section depending on the provider you are configuring, or create this section. For example, `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID`.

> ⚠ **IMPORTANT**
> Be careful when using Registry Editor. If you edit the registry incorrectly, serious problems may occur that require a complete reinstallation of the operating system or lead to data loss.

Authentication providers

Axidian Key

The policies apply to devices with the Windows Logon and ESSO Agent components installed.

Telegram

Operation timeout for the Telegram application

Software OTP

Authentication parameter settings

HOTP

One-time password synchronization and verification settings

Hardware TOTP

One-time password synchronization and verification settings

SMS OTP

Select the connection type

Storage SMS OTP

SMS sending service

Email OTP

SMTP server settings

Passcode

Password storage settings

MFA

Authentication parameter settings

Smart Card

Smart card removal behavior

IronLogic Z2USB

Smart card removal behavior

Omnikey

Smart card removal behavior

Futronic

Minimum acceptable template quality at registration

Axidian Key

> **NOTE**
>
> The policies apply to devices with the Windows Logon and ESSO Agent components installed.

Operation timeout for the Axidian Key application

The policy defines the operation waiting time for the Axidian Key application.

Require a command from the user

The policy specifies that a command from the user is required to activate the login procedure.

Telegram

Operation timeout for the Telegram application

The policy defines the operation waiting time for the Telegram application.

Require a command from the user

The policy specifies that a command from the user is required to activate the login procedure.

Software OTP

Authentication parameter settings

Before configuring the group policy, add the Axidian Access policy templates to the list of administrative templates. The policy template files are included in the provider distribution and are located in the *Misc* folder.

> ⓘ **NOTE**
> Configuring the policies is required to increase the security level. However, Software OTP Provider works properly with the default policy values as well.

- One-time password length — the policy defines the number of digits in the one-time password. You can set the value 6 or 8. If the policy is not defined, the value 6 is used by default.

  > ⚠ **IMPORTANT**
  > When using Software OTP Provider in scenarios with the Enterprise Single Sign-On and Windows Logon modules, only a six-digit password is currently supported.

- One-time password validity period — the policy defines the minimum validity period of the one-time password at registration. The period is set as an integer from 3 to 18, where 3 corresponds to a time interval of 30 seconds (+/- 15 seconds). If the policy is not defined, the value 6 is used by default.

- Minimum PIN length — the policy allows you to set the minimum number of characters the PIN must consist of. The allowed range is from 4 to 25 characters.

- Name format — the policy allows you to set a user parameter as the name of the OTP account that is transmitted in the QR code. Possible parameters:

  - CanonicalName
  - CN
  - DistiguishedName
  - FullName
  - Mail
  - PrincipalName
  - SamCompatibleName

> **NOTE**
> The policy is applied to the servers with Management Console installed. If the policy has not been applied, the username is used as the OTP account name.

Encryption algorithm settings

In Axidian Access 8.2 and higher, you can set the encryption algorithm for Software OTP Provider. If the encryption algorithm is not set, the value SHA1 is used by default. You can perform the configuration either in Group Policy Editor or in Windows Registry Editor.

In registry

1. Open Registry Editor on the machine where Management Console and User Console are deployed.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs` section and select or create the *GoogleOTP* section.
3. Create the `SecretLength` parameter of the DWORD type and specify one of the following values in the decimal system:
   - 20 — for the SHA1 algorithm
   - 32 — for the SHA256 algorithm
   - 64 — for the SHA512 algorithm

In Group Policy Editor

1. Open Group Policy Editor.
2. Go to **Computer Configuration→Administrative Templates→Axidian ID→Id Providers→SoftwareTOTP**.
3. Enable the **Secret length** policy and select the required value.

HOTP

One-time password synchronization and verification settings

Sets the counter value interval for the synchronization and for the one-time password verification.

Default values:

- Synchronization window — 2000
- One-time password verification window — 5

Hardware TOTP

One-time password synchronization and verification settings

Sets the counter value interval for the synchronization and for the one-time password verification.

Default values:

- Synchronization window — 100
- One-time password verification window — 1

Manual time configuration

Sets the correspondence between the device model and its initial time.

> **INFORMATION**
> For the "eTPass 6.20" model, by default (if the policy is not set), the time step is 30 seconds and the initial time is 1/1/2000.
>
> By default, for all other devices, the time step is 30 and the initial time is 1/1/1970.

> **INFORMATION**
> In the **Value name** field, specify the device name. In the **Value** field, specify the parameters: *Initial time*, *Step*, *PIN length*.

SMS OTP

Select the connection type

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP` section.
3. Create the *SmsSenderType* parameter of the DWORD type with one of the following values:
   - 0 — Smpp
   - 1 — Megafon
   - 2 — OneGate
   - 3 — Kafka
   - 4 — MFM Solutions
   - 5 — SMS Proxy
   - 6 — Messages Proxy

In Group Policy Editor

The policy is applied to Core Servers and is intended for selecting the Sender of the SMS OTP provider.

Not Configured or Disabled — the SMPP connection is used.

Enabled — the connection through the specified provider is used.

Service Configurations SMPP

Service Configurations SMPP

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP` section.
3. Create the following parameters:
    - `enableTls` (DWORD) — the use of encryption (1 or 0)
    - `srvAddr` (string parameter) — the address for connecting to the server
    - `srvPort` (DWORD) — the port for connecting to the server
    - `srvSystemId` (REG_SZ) — the account name for connecting to the server
    - `srvPassword` (REG_SZ) — the account password for connecting to the server
    - `srvSysType` (REG_SZ) — the PDU field of the BIND_TRANSCEIVER operation of the SMPP protocol
    - `senderName` (REG_SZ) — the sender name that is displayed when the SMS is received
    - `messageOTP` (REG_MULTI_SZ) — an arbitrary message text preceding the one-time password. By default, only the OTP is sent; for additional configuration, use the following parameters:
        - `<app>` — the name of the application that sent the authorization request
        - `<requestLocalServerTime>` — the local server time when the request was received
        - `<requestComputerDns>` — the DNS name of the computer that sent the request
        - `<requestComputerIp>` — the IP address of the computer that sent the request

- `SMPPServerTimeoutSec` (DWORD) — the waiting time for receiving the status of the sent SMS from the server
- `MessageStatusPDU` (DWORD) — the PDU in which the server sends the status of the sent message. 0 — SUBMIT_SM_RESP; 1 — DELIVER_SM
- `BindTransmitter` (DWORD). 1 — *Transmitter* — sending the `BIND_TRANSMITTER` packet at connection; 0 — *Transceiver* — sending the `BIND_TRANSCEIVER` packet at connection. When connecting as Transmitter, the server sends the status of the sent SMS message only in the `SUBMIT_SM_RESP` packet
- `sourceAddrTon` (DWORD) — the Type of Number for the source address
- `sourceAddrNpi` (DWORD) — the Numbering Plan Indicator for the source address
- `destAddrTon` (DWORD) — the Type of Number for the destination
- `destAddrNpi` (DWORD) — the Numbering Plan Indicator for the destination
- `esmClass` — specifies the Message Mode & Message Type
- `registeredDelivery` — the indicator specifying that an SMS delivery receipt or an SME acknowledgement is requested
- `dataCoding` — defines the encoding scheme of the short message user data

In Group Policy Editor

The policy is applied to Core Servers and defines the following settings for the SMPP service:

- *Use tls* — the use of encryption
- *URL (IP address)* — the address for connecting to the server
- *Port* — the port for connecting to the server
- *SystemId (Login)* — the account name for connecting to the server
- *Password* — the account password for connecting to the server
- *SystemType* — the PDU field of the `BIND_TRANSCEIVER` operation of the SMPP protocol

- *Sender* — the sender name that is displayed when the SMS is received
- *Additional text before the OTP* — an arbitrary message text preceding the one-time password. By default, only the OTP is sent; for additional configuration, use the following parameters:
  - <*app*> — the name of the application that sent the authorization request
  - <*requestLocalServerTime*> — the local server time when the request was received
  - <*requestComputerDns*> — the DNS name of the computer that sent the request
  - <*requestComputerIp*> — the IP address of the computer that sent the request

    > ⚠ **IMPORTANT**
    > The display of the OTP code is not configurable; the OTP is always displayed at the end of the message.
    >
    > To add line breaks to the message being sent, you must modify the registry at `HKLM/SOFTWARE/Policies/Axidian-ID/BSPs/SMSOTP`. Change the value of the old *messageOTP* parameter of the *REG_SZ* type to *messageOTP* of the *REG_MULTI_SZ* type.
- *SMS status waiting time* — the waiting time for receiving the status of the sent SMS from the server
- *PDU with the SMS status* — the PDU in which the server sends the status of the sent message
- *Connect as*:
  - *Transmitter* — sending the `BIND_TRANSMITTER` packet at connection
  - *Transceiver* — sending the `BIND_TRANSCEIVER` packet at connection

    When connecting as Transmitter, the server sends the status of the sent SMS message only in the `SUBMIT_SM_RESP` packet.
- *source_addr_ton* — the Type of Number for the source address
- *source_addr_npi* — the Numbering Plan Indicator for the source address

- *dest_addr_ton* — the Type of Number for the destination
- *dest_addr_npi* — the Numbering Plan Indicator for the destination
- *esm_class* — specifies the Message Mode & Message Type
- *registered_delivery* — the indicator specifying that an SMS delivery receipt or an SME acknowledgement is requested
- *data_coding* — defines the encoding scheme of the short message user data

Backup SMPP service settings

Backup SMPP service settings

In the *SMPP service settings (backup)* policy, you can configure the backup SMPP service through which SMS messages are sent if the main service is unavailable.

When sending an SMS, Core Server checks the availability of the main SMPP service. If the main service is available, the SMS is sent through the main gateway. If the main service is unavailable, the SMS is sent through the backup gateway.

The service is considered unavailable if Core Server receives the following errors:

- *SMS sending service address unavailable*
- *SMPP Error Bind*
- *SMPP Error SubmitS*

The backup SMPP service policy is configured in the same way as the main SMPP service.

If a backup service is configured, the switch to it when the main service is unavailable occurs automatically.

Service Configurations Megafon

Service Configurations Megafon

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP\Megafon` section.
3. Create the following parameters:
   - *ServerUrl* (REG_SZ) — the address for connecting to the server
   - *Login* (REG_SZ) — the account name for connecting to the server
   - *Password* (REG_SZ) — the account password for connecting to the server
   - *From* (REG_SZ) — the sender name that is displayed when the SMS is received
   - *messageOTP* (REG_SZ) — an arbitrary message text preceding the one-time password

In Group Policy Editor

The policy is applied to Core Servers and defines the settings for Megafon, an SMS sending service:

- *URL* — the address for connecting to the server
- *Login* — the account name for connecting to the server
- *Password* — the account password for connecting to the server
- *Sender* — the sender name that is displayed when the SMS is received
- *Additional text before the OTP* — an arbitrary message text preceding the one-time password

Service Configurations OneGate

Service Configurations OneGate

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP\OneGate` section.
3. Create the following parameters:
   - *ServerUrl* (REG_SZ) — the address for connecting to the server
   - *Login* (REG_SZ) — the account name for connecting to the server
   - *Password* (REG_SZ) — the account password for connecting to the server
   - *NameSender* (REG_SZ) — displayed as the sender name on the recipient device
   - *NameSystem* (REG_SZ) — the combination of *Sender system name* and *Sender name* defines the login used to address the separator
   - *messageOTP* (REG_SZ) — an arbitrary message text preceding the one-time password

In Group Policy Editor

The policy is applied to Core Servers and defines the settings for OneGate, an SMS sending service:

- *URL* — the address for connecting to the server
- *Login* — the account name for connecting to the server
- *Password* — the account password for connecting to the server
- *Sender name* — displayed as the sender name on the recipient device
- *Sender system name* — the combination of *Sender system name* and *Sender name* defines the login used to address the separator
- *Additional text before the OTP* — an arbitrary message text preceding the one-time password

Service Configurations MFMSolution

Service Configurations MFMSolution

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP\MFM` section.
3. Create the following parameters:
   - *ServerUrl* (REG_SZ) — the address for connecting to the server
   - *Login* (REG_SZ) — the account name for connecting to the server
   - *Password* (REG_SZ) — the account password for connecting to the server
   - *From* (REG_SZ) — displayed as the sender name on the recipient device
   - *MessageOTP* (REG_SZ) — an arbitrary message text preceding the one-time password

In Group Policy Editor

The policy is applied to Core Servers and defines the settings for MFMSolution, an SMS sending service:

- *URL* — the address for connecting to the server
- *Login* — the account name for connecting to the server
- *Password* — the account password for connecting to the server
- *Sender name* — displayed as the sender name on the recipient device
- *Additional text before the OTP* — an arbitrary message text preceding the one-time password

Apache Kafka service settings

Apache Kafka service settings

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP` section.
3. Create the following parameters:
   - *SmsSenderType* (DWORD) — the SMS sender type (for Apache Kafka, set the value 3)
   - *ParsePhone* (DWORD) — enables phone number parsing
   - *PhonePrefix* (REG_SZ) — the phone number prefix (if you have enabled phone number parsing)
   - *PhoneLength* (DWORD) — the phone number length (if you have enabled phone number parsing)
4. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP\Kafka` section, create the following parameters:
   - messageOTP (REG_SZ) — an arbitrary message text preceding the one-time password
   - MessageFormatPath (REG_SZ) — the path to the json file with the message format
   - MessageDateFormat (REG_SZ) — the date format in the message fields
   - ServerUrl (REG_SZ) — the Apache Kafka server address
   - MessageTimeoutSec (REG_DWORD) — the local message waiting time. This value is applied only locally and limits the time during which the message being created waits for successful delivery. If you specify the value 0 sec, the waiting time is infinite
   - TopicName (REG_SZ) — the name of the Apache Kafka topic the messages are published to
   - UseSerialization (REG_DWORD) — enables or disables serialization
   - SchemaRegistryUrl — the address for connecting to the *Schema Registry*

- RecordSchemaPath (REG_SZ) — the path to the json file with the serialization schema

If you use authorization in Apache Kafka, add the following parameters:

- SecurityProtocol (REG_SZ) — the protocol used for communication with brokers
- SaslMechanism (REG_SZ) — the SASL mechanism for authentication

In Group Policy Editor

The policy is applied to Core Servers and defines the settings for Apache Kafka, an SMS sending service:

- *URL* — the address for connecting to the server
- *Server timeout* — the local message waiting time. This value is applied only locally and limits the time during which the message being created waits for successful delivery. If you specify the value 0 sec, the waiting time is infinite
- *Enable message serialization* — the flag enables or disables message serialization
- *Schema registry URL* — the address for connecting to the *Schema Registry*
- *Path to the serialization schema* — the path to the json file with the serialization schema
- *Security protocol* — the protocol used for communication with brokers
- *SASL mechanism* — the SASL mechanism for authentication
- *Topic name* — the name of the topic the messages are published to
- *Message date format* — the date format in the message fields
- *Additional text before the OTP* — an arbitrary message text preceding the one-time password
- *Path to the message format* — the path to the json file with the message format

SMS Proxy settings

SMS Proxy settings

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP\SMSProxy` section.
3. Create the following parameters:
   - *ServerUrl* — the address for connecting to SMS Proxy
   - *From* (REG_SZ) — the sender name that is displayed when the message is received
   - *messageOTP* (REG_SZ) — an arbitrary message text preceding the one-time password
   - *ClientCertificateThumbprint* (REG_SZ) — defines the certificate that is added to the client certificate list when requesting SMS Proxy (the certificate itself must be located in the LocalMachine/LocalComputer certificate store and meet the SMS Proxy requirements)
   - *ServerCertificateThumbprint* (REG_SZ) — defines the server certificate that is recognized as valid

In Group Policy Editor

The policy is applied to Core Servers:

- *URL* — the address for connecting to SMS Proxy
- *Sender* — the sender name that is displayed when the SMS is received
- *Additional text before sending the OTP* — an arbitrary message text preceding the one-time password
- *Client certificate thumbprint* — defines the certificate that is added to the client certificate list when requesting SMS Proxy (the certificate itself must be located in the LocalMachine/LocalComputer certificate store and meet the SMS Proxy requirements)
- *Server certificate thumbprint* — defines the server certificate that is recognized as valid

Messages Proxy settings

Messages Proxy settings

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP\MessagesProxy` section.
3. Create the following parameters:

- *ServerUrl* — the address for connecting to the server
- *From* (REG_SZ) — the sender name that is displayed when the message is received
- *messageOTP* (REG_SZ) — an arbitrary message text preceding the one-time password
- *ClientCertificateThumbprint* (REG_SZ) — defines the certificate that is added to the client certificate list when requesting Messages Proxy (the certificate itself must be located in the LocalMachine/LocalComputer certificate store and meet the Messages Proxy requirements)
- *ServerCertificateThumbprint* (REG_SZ) — defines the server certificate that is recognized as valid

In Group Policy Editor

The policy is applied to Core Servers:

- *URL* — the address for connecting to the server
- *Sender* — the sender name that is displayed when the message is received
- *Additional text before sending the OTP* — an arbitrary message text preceding the one-time password
- *Client certificate thumbprint* — defines the certificate that is added to the client certificate list when requesting Messages Proxy (the certificate itself must be located in the LocalMachine/LocalComputer certificate store and meet the Messages Proxy requirements)
- *Server certificate thumbprint* — defines the server certificate that is recognized as valid

Message format settings

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP` section.
3. Create the following parameters:
   - *SMSOTP* (DWORD) with the value 1
   - *dateFormat* (REG_SZ) — allows you to set the date display format. You can find format examples in the Microsoft article

In Group Policy Editor

The policy allows you to set the date display format. You can find format examples in the Microsoft article.

Enabled — the date is displayed according to the format specified in the policy.

One-time password settings

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP` section.
3. Create the following parameters:
   - *OTPGeerator* (DWORD) — enables this setting
   - *OneTimePasswordLength* (DWORD) — the one-time password length
   - *AllowDigitsChGr* (DWORD) — use digits in the OTP
   - *AllowLatinsLowerChGr* (DWORD) — use lowercase Latin letters in the OTP
   - *AllowLatinsUpperChGr* (DWORD) — use uppercase Latin letters in the OTP
   - *AllowSpecialChGr* (DWORD) — use special characters in the OTP

In Group Policy Editor

The policy is applied to Core Servers and allows you to set the length and the character group occurrence when generating the one-time password.

- *Not Configured* — the one-time password consists of digits and lowercase Latin letters and is 6 characters long.
- *Enabled* — the one-time password is generated according to the rules specified in the policy. The password length must be at least 4 and no more than 24 characters.
- *Disabled* — the one-time password consists of digits and lowercase Latin letters and is 6 characters long.

Phone number format settings

In registry

1. Open Registry Editor on the Core Server.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\SMSOTP` section.
3. Create the following parameters:
   - *ParsePhone* (DWORD) — enables phone number parsing
   - *PhoneLength* (DWORD) — the phone number length
   - *PhonePrefix* (string parameter) — the phone number prefix

In Group Policy Editor

The policy is applied to Core Servers and allows you to configure the phone number format. Enabling this policy enables phone number normalization. This means the following:

- Spaces, tabs, brackets, and hyphens are removed.
- If several numbers are written in the string and separated by a comma or a semicolon, the first number in the string up to the separator is taken.
- 8 is replaced with +7.
- If +7 or 8 is missing at the beginning of the string, +7 is added before the number.
- If the letter O is specified in the number instead of the digit 0, it is replaced back with the digit.

> **NOTE**
>
> Due to the REST request format requirements, the phone number for Megafon must be of the long type. The **+** sign is not taken into account during normalization.

Concurrent connection to the SMPP server settings

The policy is applied to Core Servers and defines the processing order of requests to the SMPP server. Enabling the policy may be necessary if the SMPP server does not support several simultaneous connections from one user (the account specified in the *SMPP service settings* policy).

Not Configured or Disabled — the connection to the SMPP server and the message sending requests are performed in parallel.

Enabled — the connection to the SMPP server and the message sending requests are performed sequentially.

Storage SMS OTP

SMS sending service

The policy is applied to Core Servers and defines the following settings for the SMS sending server:

- *Use tls* — the use of encryption
- *URL (IP address)* — the address for connecting to the server
- *Port* — the port for connecting to the server
- *SystemId (Login)* — the account name for connecting to the server
- *Password* — the account password for connecting to the server
- *SystemType* — the PDU field of the `BIND_TRANSCEIVER` operation of the SMPP protocol
- *Sender* — the sender name that is displayed when the SMS is received
- *Additional text before the OTP* — an arbitrary message text preceding the one-time password. By default, only the one-time password is sent; for additional configuration, use the following parameters:
  - *<app>* — the name of the application that sent the authorization request
  - *<requestLocalServerTime>* — the local server time when the request was received
  - *<requestComputerDns>* — the DNS name of the computer that sent the request
  - *<requestComputerIp>* — the IP address of the computer that sent the request

> ⓘ **INFORMATION**
> The display of the one-time password is not configurable; the password is always displayed at the end of the message.
>
> To add line breaks to the message being sent, modify the registry at `HKLM/SOFTWARE/Policies/Axidian-ID/BSPs/SMSOTP`. Change the value of the old *messageOTP* parameter of the *REG_SZ* type to *messageOTP* of the *REG_MULTI_SZ* type.

- *SMS status waiting time* — the waiting time for receiving the status of the sent SMS from the server
- *PDU with the SMS status* — the PDU in which the server sends the status of the sent message
- *source_addr_ton* — the Type of Number for the source address
- *source_addr_npi* — the Numbering Plan Indicator for the source address
- *dest_addr_ton* — the Type of Number for the destination
- *dest_addr_npi* — the Numbering Plan Indicator for the destination
- *esm_class* — specifies the Message Mode & Message Type
- *registered_delivery* — the indicator specifies that an SMS delivery receipt or an SME acknowledgement is requested
- *data_coding* — defines the encoding scheme of the short message user data

Message format settings

The policy allows you to set the date display format. You can find format examples in the Microsoft article.

Enabled — the date is displayed according to the format specified in the policy.

One-time password settings

The policy is applied to Core Servers and allows you to set the length and the character group occurrence when generating the one-time password.

- *Not Configured* — the one-time password consists of digits and lowercase Latin letters and is 6 characters long.
- *Enabled* — the one-time password is generated according to the rules specified in the policy. The password length must be at least 4 and no more than 24 characters.
- *Disabled* — the one-time password consists of digits and lowercase Latin letters and is 6 characters long.

Concurrent connection to the SMPP server settings

The policy is applied to Core Servers and defines the processing order of requests to the SMPP server. Enabling the policy may be necessary if the SMPP server does not support several simultaneous connections from one user (the account specified in the *SMS sending service* policy).

Not Configured or Disabled — the connection to the SMPP server and the message sending requests are performed in parallel.

Enabled — the connection to the SMPP server and the message sending requests are performed sequentially.

Email OTP

SMTP server settings

The policy is applied to Core Servers and allows you to set the following parameters for working with an SMTP server:

- *Server (DNS name, IP address)* — the address for connecting to the server
- *Port* — the port for connecting to the server
- *Server timeout* — the response waiting time from the server, in seconds
- *Connection type* — unsecured, TLS, or SSL
- *Login* — the account name for connecting to the server
- *Password* — the account password for connecting to the server
- *One-time password in the message subject* — if the option is enabled, the one-time password is displayed in the message subject; otherwise, it is displayed in the message body
- *Message text* — the sender name and email, the message subject and body

> ⓘ **NOTE**
> In the message text, you must specify the insertion point for the one-time password with the <`otp`> tag.
>
> If the *One-time password in the message subject* option is set, the same insertion point for the one-time password must be specified in the message subject.

Not Configured or Disabled — if the policy is not defined or is disabled, Email OTP Provider is not used for user authentication.

Enabled — when the policy is enabled, Email OTP Provider works according to the parameters defined in the policy.

Passcode

Password storage settings

The policy enables storing the hash sum of the Passcode password instead of storing its value. Hashing is performed with the SHA-1 algorithm.

If the policy is not defined or is disabled, the password is stored in its original form.

Password requirement settings

> ⓘ **NOTE**
> For the changes in the policy settings to take effect, you must update the group policy. To update the group policy immediately, use the `gpupdate /force` command.

The policy allows you to set the following requirements for the Passcode password:

- The minimum password length
- The maximum number of occurrences of each character
- The allowed characters (Latin letters, digits, special characters, custom characters)
- The minimum number of characters from a group

The maximum number of occurrences of each character can be from 1 to 63, or 0.

Character groups:

- Digits: 0-9
- Lowercase Latin letters: a-z
- Uppercase Latin letters: A-Z
- Special characters: . , < > / ? [ ] { } = + - _ \ | ! @ # $ % ^ & * ( )
- Custom character group (the character group is defined by the user for each application)

> ⓘ **INFORMATION**
> If the policy is not defined or is disabled, the password can consist only of digits and lowercase Latin letters and must contain at least 6 characters.

MFA

Authentication parameter settings

> ⚠ **IMPORTANT**
> The policy must be applied to all Core Servers and to all client machines.

In registry

1. Open Registry Editor.
2. Open the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\BSPs\MFA` section.
3. Create the `BSPChain` parameter of the `REG_MULTI_SZ` type and specify the identifiers of the authentication providers to be used in the chain. Specify each identifier in curly brackets and on a new line, without spaces or other characters.

List of supported providers for Windows Logon and Enterprise SSO

```
SMS OTP {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}
Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
Software OTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
Windows Password {CF189AF5-01C5-469D-A859-A8F2F41ED153}
Z2 USB {CB5109DA-B575-422C-8805-524FE12B02F5}
Futronic {A0EF00AD-1EEB-4D48-8BCF-06E19CD5585F}
smart card or USB token {0AF65AD8-DB77-4B64-B489-958D9B36E28C}
HID OMNIKEY {4B15AF52-A795-4CA6-B7CD-CDB8ABF2D2C2}
Axidian Key {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}
Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
Secured TOTP {F15FD7EC-19EA-4384-846E-A2D0BE149FA2}
```

List of supported providers for NPS RADIUS Extension

```
Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
Hardware TOTP {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
Secured TOTP {F15FD7EC-19EA-4384-846E-A2D0BE149FA2}
Software TOTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
Windows Password {CF189AF5-01C5-469D-A859-A8F2F41ED153}
Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
```

4. Create the `MFADeviceName` parameter of the `REG_SZ` type and specify the name for the created chain.

> **INFORMATION**
> This value is displayed in the name of the MFA device for the user and in the system events.

In Group Policy Editor

1. Open the **Multifactor authentication chain settings** policy. The policy is located at *Administrative Templates→Axidian-ID→Id Providers→MFA*.
2. Set the policy value to *Enabled*.
3. In the *Multifactor authentication chain* parameter, specify the identifiers of the authentication providers to be used in the chain. Specify each identifier in curly brackets and on a new line, without spaces or other characters.

> **NOTE**
> A provider that is prohibited from use can be used in the MFA provider chain.

List of supported providers for Windows Logon and Enterprise SSO

```
SMS OTP {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}
Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
Software OTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
Windows Password {CF189AF5-01C5-469D-A859-A8F2F41ED153}
Z2 USB {CB5109DA-B575-422C-8805-524FE12B02F5}
Futronic {A0EF00AD-1EEB-4D48-8BCF-06E19CD5585F}
smart card or USB token {0AF65AD8-DB77-4B64-B489-958D9B36E28C}
HID OMNIKEY {4B15AF52-A795-4CA6-B7CD-CDB8ABF2D2C2}
Axidian Key {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}
Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
Hardware TOTP {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
Secured TOTP {F15FD7EC-19EA-4384-846E-A2D0BE149FA2}
```

List of supported providers for NPS RADIUS Extension

```
Hardware OTP {AD3FBA95-AE99-4773-93A3-6530A29C7556}
Hardware TOTP {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
Secured TOTP {F15FD7EC-19EA-4384-846E-A2D0BE149FA2}
Software TOTP {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
Windows Password {CF189AF5-01C5-469D-A859-A8F2F41ED153}
Passcode {F696F05D-5466-42b4-BF52-21BEE1CB9529}
```

Example of the Passcode + SMS OTP chain

{F696F05D-5466-42b4-BF52-21BEE1CB9529}

{EBB6F3FA-A400-45F4-853A-D517D89AC2A3}

4. In the *Device name* parameter, specify the name for the created chain. The default value is *MFA*.

> **INFORMATION**
> This value is displayed in the name of the MFA device for the user and in the system events.

RADIUS Extension policy settings

When integrating with the RADIUS Extension module, the MFA authentication method is called *One-string authenticator*. With this multifactor authentication method, users are provided with a single field for entering all factors. Each authentication factor is entered into the input field in one line, with or without a separator.

OTP length

Allows you to set the one-time password length if no separator is used in the provider chain. If the policy is enabled, the one-time password is separated from the permanent password by defining the number of characters in the password. The specified length is applied to all one-time password providers in the chain.

In registry

1. Open Registry Editor.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius\MFA` section.
3. Create the `OTPLen` parameter of the `REG_DWORD` type and specify the value that matches the value in the One-time password length policy (the value 6 or 8).

In Group Policy Editor

1. Open the *OTP length* policy. The policy is located at *Administrative Templates→RADIUS→MFA*.
2. Set the policy value to *Enabled*.
3. Set the value that matches the value in the One-time password length policy (the value 6 or 8).

Separator character

Allows you to set the character that separates the authenticators. Any single character can be used as a separator: digits, lowercase and uppercase Latin and Cyrillic letters, special characters (including the space).

> ⚠ **IMPORTANT**
> If the authenticator already contains the separator character, the user must duplicate the separator character when entering it in the form. For example, the authenticator is password\example and the one-time password is 395816. The following must be entered in the input form: password\\example\395816.

> ⓘ **INFORMATION**
> If both the *OTP length* and the *Separator character* policies are enabled, the *Separator character* policy takes priority.

In registry

1. Open Registry Editor.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius\MFA` section.
3. Create the `Separator` parameter of the `REG_SZ` type and specify the character that separates the permanent password and the one-time password.

In Group Policy Editor

1. Open the *Separator character* policy. The policy is located at *Administrative Templates→RADIUS→MFA*.
2. Set the policy value to *Enabled*.
3. Set the character that separates the permanent password and the one-time password.

Support for smart card policies

If the chain contains providers such as Smart Card Provider, IronLogic Z2USB Provider, or OMNIKEY Provider, the *Smart card removal behavior* and *Timeout of the action performed on smart card removal* policies are supported.

Smart Card

Smart card removal behavior

Smart Card Provider supports the Windows security policy *Interactive logon: Smart card removal behavior*.

With this policy, you can define the action performed when a smart card is removed from the user workstation.

Timeout of the action performed on smart card removal

The policy of the Windows Logon section defines the duration of the standard and the service timeout after the authentication device is removed.

The policy sets the time interval in seconds between the smart card removal and the action performed according to the Windows *Interactive logon: Smart card enhanced removal behavior* policy.

> ⓘ **INFORMATION**
> The `AxidianID.Client.admx` policy administrative template file is included in the distribution and is located in the *Misc* directory.

The standard timeout prevents the computer from being locked automatically if the authentication device is removed accidentally.

The service timeout prevents the computer from being locked automatically in cases when removing the authentication device in use is necessary (registering an additional authenticator, accessing the operating system or an application under another account and with another authenticator). To activate the service timeout, press and hold the `[Ctrl]+[L]` key combination before removing the device.

If the policy is not set or is disabled, no timeout is provided before the workstation is locked automatically.

Use extra data

The policy allows you to configure the enhanced security mode with the ability to write extra data to the device and read this data. The extra data mode prevents the device from being re-initialized and

the authenticator from being subsequently used without authorization on behalf of the account of the actual owner.

> **INFORMATION**
>
> The `AxidianID.Smartcard.Provider.admx` policy administrative template file is included in the distribution and is located in the *Misc* directory.

Extra data is a sequence of 64 random bytes that is written to the smart card (USB token) and included in the authenticator along with the serial number of the authentication device. The extra data is read from the device and verified together with the serial number during user authentication.

The extra data mode is enabled if the policy setting is set to *Enabled* or *Not Configured*. Allow operation without extra data — allows the provider to work without using extra data (when the policy is enabled).

The option can be used to work with devices registered in older versions of Smart Card Provider (without extra data support).

Waiting for the smart card after logging in to a terminal session

The policy allows you to set the time in seconds during which the smart card is awaited after logging in to a terminal session in case of incorrect operation of the PKCS smart card.

IronLogic Z2USB

Smart card removal behavior

IronLogic Z-2 USB Provider supports the Microsoft security policy *Interactive logon: Smart card removal behavior*.

With this policy, you can define the action performed when a smart card is removed from the user workstation.

Timeout of the action performed on smart card removal

> **NOTE**
> Configuring the policies is required to increase the security level; however, IronLogic Z-2 USB Provider works properly with the default policy values as well.

The policy of the Windows Logon section defines the duration of the standard and the service timeout after the authentication device is removed.

The policy sets the time interval in seconds between the smart card removal and the action performed according to the Windows *Interactive logon: Smart card enhanced removal behavior* policy.

The standard timeout prevents the computer from being locked automatically if the authentication device is removed accidentally.

The service timeout prevents the computer from being locked automatically in cases when removing the authentication device in use is necessary (registering an additional authenticator, accessing the system on behalf of another account with another authenticator, and so on). To activate the service timeout, press and hold the `[Ctrl]+[L]` key combination before removing the device.

If the policy is not set or is disabled, no timeout is provided before the workstation is locked automatically.

Use the contactless card identifier in the ACS format

The policy defines the storage format of card identifiers in the Axidian Access database. This makes it possible to ensure compatibility with the identifier storage format of access control systems (ACS).

By default, the full card information is stored:

- Type
- Manufacturer
- Batch number and card identifier

Omnikey

Smart card removal behavior

OMNIKEY Provider supports the Windows security policy *Interactive logon: Smart card removal behavior*.

With this policy, you can define the action performed when a smart card is removed from the user workstation.

Timeout of the action performed on smart card removal

The policy of the Windows Logon section defines the duration of the standard and the service timeout after the authentication device is removed.

The policy sets the time interval in seconds between the smart card removal and the action performed according to the Windows *Interactive logon: Smart card enhanced removal behavior* policy.

The standard timeout prevents the computer from being locked automatically if the authentication device is removed accidentally.

The service timeout prevents the computer from being locked automatically in cases when removing the authentication device in use is necessary (registering an additional authenticator, accessing the system on behalf of another account with another authenticator, and so on). To activate the service timeout, press and hold the `[Ctrl]+[L]` key combination before removing the device.

If the policy is not set or is disabled, no timeout is provided before the workstation is locked automatically.

Use PACS data

> ⓘ **INFORMATION**
> The policy is applied to user workstations.

When enabled, this policy allows using PACS data instead of the card identifier.

It requires no configuration.

Futronic

Minimum acceptable template quality at registration

The policy defines the minimum acceptable quality of the fingerprint template at registration. The template quality is set as an integer from 1 to 10, where 10 corresponds to the maximum possible template quality.

> ⓘ **NOTE**
> The policy must be defined on the user workstations where the client components of the system are installed. If the policy is not defined, the value *3* is used by default.

To achieve the maximum template quality at registration, place the finger on the scanner carefully and do not change the finger position relative to the scanner.

Maximum acceptable FAR

The policy defines the maximum acceptable FAR (False Acceptance Rate) value when comparing fingerprint templates.

> ⓘ **NOTE**
> The policy must be defined on all Core Servers. If the policy is not defined, the default value is *250*.

It is set as an integer from 1 to 1000. The value 1 corresponds to the most "lenient" comparison, when the probability of comparing different templates and getting a positive result is maximum. The value 1000 corresponds to the most "strict" comparison, when such probability is minimum.

Setting the parameter value too high may lead to frequent errors related to denying access to legitimate users.

Scanner indication settings

The policy sets the operation parameters of the scanner LEDs. By default, the scanner LEDs do not blink in either mode.

RADIUS Extension

The policies are applied to the servers with the NPS (Network Policy Server) role deployed and allow you to perform additional settings.

You can configure the policies either with domain group policies or with the local group policy on the NPS server. After configuring the policies, you must restart the NPS service.

Configure the Challenge\Response session timeout

Allows you to set the session timeout when using an authentication provider that supports Challenge\Response.

In registry

1. Open Registry Editor on the NPS server.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius` section.
3. Create a parameter of the DWORD type named `SessionLifetimeSec`. As the value, specify the session timeout in seconds in the decimal format.

   Example

   ```
   Windows Registry Editor Version 5.00

   [HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius]
   "SessionLifetimeSec"=dword:0000001e
   ```

In Group Policy Editor

1. Open Group Policy Editor.
2. Go to **Computer Configuration→Administrative Templates→Axidian ID→Radius**.
3. Open the *Challenge\Response session timeout settings* policy.
4. Enable the policy. In the **Session timeout in seconds** field, specify the required value.

Challenge\Response: message to the user

The policy allows you to set the message shown to the user when Challenge\Response is used.

In registry

1. Open Registry Editor on the NPS server.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius` section.
3. Create a section with the name of the required provider: *EmailOTP*, *HardwareHOTP*, *GoogleOTP*, *SMSOTP*.
4. Create a string parameter with a name in the format *<name of the provider used>ChallengeResponseReplyMessage*, such as *HardwareHOTPChallengeResponseReplyMessage*.
5. As the value, specify the text that is displayed to the user.

Example

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius\EmailOTP]
"EmailOTPChallengeResponseReplyMessage"="EmailOTP:"

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius\HardwareHOTP]
"HardwareHOTPChallengeResponseReplyMessage"="HardwareHOTP: "

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius\GoogleOTP]
"GoogleOTPChallengeResponseReplyMessage"="Software TOTP OTP: "

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius\SMSOTP]
"SMSOTPChallengeResponseReplyMessage"="SMS OTP: "
```

In Group Policy Editor

1. Open Group Policy Editor.
2. Go to **Computer Configuration→Administrative Templates→Axidian ID→Radius→<Name of the provider used>**.
3. Open the *Challenge\Response message to the user* policy.

4. Enable the policy. In the **Message to the user** field, enter the required text.

Detect the RADIUS client IP address

The policy allows you to set the attributes for detecting the RADIUS client IP address that is used to search for the RADIUS application in Axidian Access. If no value is assigned, the default value corresponds to 265 (SrcIPAddress), 278 (SrcIPv6Address).

In registry

1. Open Registry Editor on the NPS server.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius` section.
3. Create a parameter of the REG_SZ type named `ClientIPAttributes` and set the required value (you can specify several values separated by commas):
   - 4 — NASIPAddress
   - 95 — NASIPv6Address
   - 265 — SrcIPAddress
   - 278 — SrcIPv6Address

In Group Policy Editor

1. Open Group Policy Editor.
2. Go to **Computer Configuration→Administrative Templates→Axidian ID→Radius**.
3. Open the *Detect the RADIUS client IP address* policy.
4. Enable the policy. In the **Attributes** field, specify the required value. You can specify several values separated by commas.

User group caching

The policy enables user group caching during RADIUS authentication and allows you to set the cache update period.

> **NOTE**
> If NPS RADIUS Extension or Core Server respond slowly when working under load with a large number of users, you must disable this setting.

In registry

1. Open Registry Editor on the NPS server.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius` section.
3. Create a parameter of the DWORD type named `UserGroupsCachingEnabled` and set the parameter value to 1.
4. Create a parameter of the DWORD type named `UserGroupsCacheUpdateMin` and set the parameter value to a decimal number in minutes.

Example

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius]
"UserGroupsCachingEnabled"=dword:00000001
"UserGroupsCacheUpdateMin"=dword:00000021
```

In Group Policy Editor

1. Open Group Policy Editor.
2. Go to **Computer Configuration→Administrative Templates→Axidian ID→Radius**.
3. Open the **User group caching settings** policy.
4. Enable the policy. In the **User group update period in minutes** field, specify the required value in minutes.

User name settings

The policy allows you to configure the use of the NetBIOS domain name when the username is specified without the domain.

In registry

1. Open Registry Editor on the NPS server.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius` section.
3. Create a parameter of the DWORD type named `UseNetBiosDomainName` with the value *1*.

Example

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius]
"UseNetBiosDomainName"=dword:00000001
```

In Group Policy Editor

1. Open Group Policy Editor.
2. Go to **Computer Configuration→Administrative Templates→Axidian ID→Radius**.
3. Open the *User name settings* policy.
4. Enable the policy and activate the **Use the NetBIOS domain name when the username is specified without the domain** parameter.

User request session caching settings

The policy enables the caching of user request sessions during RADIUS authentication and allows you to set the request session lifetime in seconds.

In registry

1. Open Registry Editor on the NPS server.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius` section.
3. Create a parameter of the DWORD type named `RequestSessionCachingEnabled` and set the parameter value to *1*.

4. Create a parameter of the DWORD type named `RequestSessionLifetimeSec` and set the parameter value to a decimal number in seconds.

Example

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius]
"RequestSessionCachingEnabled"=dword:00000001
"RequestSessionLifetimeSec"=dword:0000003c
```

In Group Policy Editor

1. Open Group Policy Editor.
2. Go to **Computer Configuration→Administrative Templates→Axidian ID→Radius**.
3. Open the *User request session caching settings* policy.
4. Enable the policy. In the **Request session lifetime in seconds** field, specify the value.

Disable the use of the login method settings from Management Console

In Axidian Access 8.2 and higher, you can configure the login method in the Management Console application. In version 8.1 and earlier, the login method was configured with the *Login method settings for user groups* and *Common login method settings* policies.

You can use your existing login method settings in the current version of Axidian Access. To do this, configure the *Disable the use of Axidian Access policies* policy.

In registry

1. Open Registry Editor on the NPS server.
2. Open or create the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Radius` section.
3. Create a parameter of the DWORD type named `DisableAccessManagerPolicies` and set the parameter value to *1*.

In Group Policy Editor

1. Open Group Policy Editor.
2. Go to **Computer Configuration→Administrative Templates→Axidian ID→Radius**.
3. Open the *Disable the use of Axidian Access policies* policy.
4. Enable the policy.

RDP Windows Logon

Configure concurrent operation with the Windows Logon module

If the scenario implies installing RDP Windows Logon and Windows Logon on the same machine, you must configure the policy for Windows Logon.

In registry

1. Go to the machine with the RDP Windows Logon and Windows Logon components installed.
2. Open Registry Editor.
3. Go to **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Logon for Windows**.
4. Create a `DWORD` parameter named `CredProvFilter` and set the value *2*.

In Group Policy Editor

1. Go to the machine with the RDP Windows Logon and Windows Logon modules installed.
2. Open Group Policy Editor.
3. Go to **Computer Configuration→Administrative Templates→Axidian ID→Windows Logon**.
4. Enable the **Credential Provider settings** policy and set the **Display of login methods** parameter to *All except the password*.

Enterprise Single Sign-On

Configure the connection to Core Server

In registry

1. Open Windows Registry Editor.
2. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2`.
3. Modify the `ServerUrlBase` string parameter and specify the URL of your Core Server, such as `http(s)://dc.axidian.local/am/core/`.

> **NOTE**
> When using the HTTPS protocol connection, you must install a client certificate on each Core Server.

In Group Policy Editor

If you configure the server connection with policies, the values are specified in the registry path `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\SrvLocator2` and take priority over the manual configuration.

1. Add the `AxidianID.ServerUrl.admx` policy to the workstation with ESSO Agent installed.
2. In *gpedit.msc*, go to **Computer Configuration→Administrative Templates→Axidian ID→ClientConnection→Server connection settings** and enable the policy.
3. In the **Core Server URL** field, specify the URL of your Core Server, such as `http(s)://dc.axidian.local/am/core/`.

Core Server availability check

For client computers, you can enable Core Server availability check, as well as to configure the time period after which Core Server becomes unavailable and the client computer starts using cached data.

By default and when the policy is disabled, Core Server availability checks are turned off and not performed, and the time for determining server unavailability is unlimited.

When configuring with policies, the values are specified in the registry path `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\AM\ServerAvailability` and take priority over the manual configuration.

To configure the Core Server availability check:

In registry

1. Go to the client machine with ESSO Agent installed.
2. Open Registry Editor.
3. Go to **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\AM\ServerAvailability**.
4. Set the values of the following parameters:
   - For the `IsEnabled` parameter, set the value *1* to enable the check.
   - For the `CheckWaitTimeoutMs` parameter, set the period during which the client module waits for the server availability check result, after which the server is considered unavailable. The default value is 5000 ms.
   - For the `CheckResultTtlMs` parameter, set the time during which the result of the previous check remains valid and is used instead of performing new checks. The default value is 6000 ms.

In Group Policy Editor

1. Go to the machine with Core Server and the ESSO Agent module installed.
2. Open Group Policy Editor.
3. Go to **Computer Configuration→Administrative Templates→Axidian ID→Client Connection**.
4. Enable the **Axidian EA\ESSO server availability check** policy.
5. Set the values of the following parameters:

- For the **Check result waiting timeout** parameter, set the period during which the client module waits for the server availability check result, after which the server is considered unavailable. The default value is 5000 ms.
- For the **Check result lifetime** parameter, set the time during which the result of the previous check remains valid and is used instead of performing new checks. The default value is 6000 ms.

Windows Logon

Configure Windows Logon

In registry

1. Open Windows Registry Editor.
2. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\SrvLocator2`.
3. Modify the `ServerUrlBase` string parameter and specify the URL of your Core Server, such as `http(s)://dc.axidian.local/am/core/`.

> ⓘ **NOTE**
> When using the HTTPS protocol connection, you must install a client certificate on each Core Server.

In Group Policy Editor

> ⚠ **IMPORTANT**
> When configuring with policies, the values are specified in the registry path `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\SrvLocator2` and take priority over the manual configuration.

1. Add the `AxidianID.ServerUrl.admx` policy to the workstation with Windows Logon installed.
2. In *gpedit.msc*, go to **Computer Configuration→Administrative Templates→Axidian ID→ClientConnection→Server connection settings**.

   Enable the policy.
3. In the **Core Server URL** field, specify the URL of your Core Server, such as `http(s)://dc.axidian.local/am/core/`.

Configure concurrent operation with RDP Windows Logon

If the scenario implies installing Windows Logon and RDP Windows Logon on the same machine, you must configure the policy for Windows Logon.

In registry

1. Go to the machine with the RDP Windows Logon and Windows Logon modules installed.
2. Open Registry Editor.
3. Go to the **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\Logon for Windows** section.
4. Create a parameter of the *DWORD* type named `CredProvFilter` and set it to the value *2*.

In Group Policy Editor

1. Go to the machine with the RDP Windows Logon and Windows Logon modules installed.
2. Open Group Policy Editor.
3. Go to the **Computer Configuration→Administrative Templates→Axidian ID→Windows Logon** section.
4. Enable the *Credential Provider settings* policy and set the **Display of login methods** parameter to **All except the password**.

Core Server availability check

For client computers, you can enable Core Server availability check, as well as to configure the time period after which Core Server becomes unavailable and the client computer starts using cached data.

By default and when the policy is disabled, Core Server availability checks are turned off and not performed, and the time for determining server unavailability is unlimited.

When configuring with policies, the values are specified in the registry path `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\ServerAvailability` and take priority over the manual configuration.

To configure the Core Server availability check:

In registry

1. Go to the client machine with Windows Logon installed.
2. Open Registry Editor.
3. Go to **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID\ServerAvailability**.
4. Set the values of the following parameters:
   - For the `IsEnabled` parameter, set the value *1* to enable the check.
   - For the `CheckWaitTimeoutMs` parameter, set the period during which the client module waits for the server availability check result, after which the server is considered unavailable. The default value is 5000 ms.
   - For the `CheckResultTtlMs` parameter, set the time during which the result of the previous check remains valid and is used instead of performing new checks. The default value is 6000 ms.

In Group Policy Editor

1. Go to the machine with Core Server and the Windows Logon module installed.
2. Open Group Policy Editor.
3. Go to **Computer Configuration→Administrative Templates→Axidian ID→Client Connection**.
4. Enable the **Axidian EA\ESSO server availability check** policy.
5. Set the values of the following parameters:
   - For the **Check result waiting timeout** parameter, set the period during which the client module waits for the server availability check result, after which the server is considered unavailable. The default value is 5000 ms.

- For the **Check result lifetime** parameter, set the time during which the result of the previous check remains valid and is used instead of performing new checks. The default value is 6000 ms.

Backup

We recommend that you prepare backup copies of the database, the configuration files, and the registry settings after you have deployed Axidian Access and verified that it works correctly.

The backup frequency depends on how you operate Axidian Access. For example, we recommend that you create backup copies of the configuration files if the parameters in the configuration files have been changed, and monthly after that.

We recommend that you create backup copies of the Axidian Access database after the deployment and debugging period. During the period when users register authenticators, we recommend that you create database backup copies daily, because the database changes when policies and authenticator settings are modified, and when users register or modify authenticators.

We recommend that you create backup copies of the following components:

Databases

- The Core Server database
- The Key Server database (if any)
- The Log Server event database (if any)

Component configuration files

- Log Server configuration files:
  - `C:\inetpub\wwwroot\ls\clientApps.config`
  - `C:\inetpub\wwwroot\ls\targetConfigs\sampleDb.config`, if the logs are stored in Microsoft SQL
  - `C:\inetpub\wwwroot\ls\targetConfigs\postgresDb.config`, if the logs are stored in PostgreSQL
  - `C:\inetpub\wwwroot\ls\targetConfigs\sampleSyslog.config`, if the logs are stored in SysLog

> **NOTE**
>
> Before creating a backup copy, decrypt the configuration file with the `EA.Config.Encryptor.exe` utility (located in the `Axidian Access Manager Server/<version number>/Misc/EA.Config.Encryptor` distribution directory). For detailed information about decryption, see the Manual Core Server configuration section. If backup databases were used for the logs, save all the configured files.

- The administrator console configuration file `C:\inetpub\wwwroot\am\mc\Web.config`
- The User Console configuration file (if any) `C:\inetpub\wwwroot\am\uc\Web.config`
- The Identity Provider configuration file (if any):
  - For Axidian Access 8.1.4 and lower, the file to back up is `C:\inetpub\wwwroot\am\idp\Web.config`
  - For Axidian Access 8.1.5 and higher, the file to back up is `C:\inetpub\wwwroot\am\idp\app-settings.json`
- The Key Server configuration file (if any):
  - For Axidian Access 8.1.x and lower, the file to back up is `C:\inetpub\wwwroot\airkeycloud\Web.config`
  - For Axidian Access 8.2 and higher, the file to back up is `C:\inetpub\wwwroot\axidiankey\Web.config`

Registry settings

- The registry settings from Core Server located in the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID` section
- The registry settings from the servers with the integration modules installed, located in the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID` and `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID` sections
- The registry settings from the client machines where the Windows Logon and/or ESSO Agent modules are used, located in the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID` and `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Axidian-ID` sections

Configuration wizard settings

To create a backup copy of the configuration wizard settings, run the configuration wizard (located in the `Axidian Access Manager Server/<version number>/Misc/EA.Server.Wizard.exe` distribution directory). Go to the **Confirmation** step, make sure that the **Save a backup copy of the configuration parameters** option is selected, and click **Apply**. Save the created backup copy.

Administration

Applications

3 items

Licenses

Managing licenses

Authenticators

Authenticator general settings in Management Console

Users

Managing users in Axidian Access

Policies

Managing user access and permissions

Devices

Managing authentication devices

Events

Managing user events

Configuration

Authenticator settings and access rights

Applications

This section includes applications and modules with registered licenses. Also, in this section, you can manually add applications integrated with ESSO Agent and NPS Radius Extension.

> **NOTE**
> Before adding an application, register licenses for the corresponding module.

Other integration modules are automatically displayed in the **Applications** section after you register licenses.

Enterprise SSO Extension

In the **Applications** section, you can add and configure applications integrated with the ESSO module, as well as set additional settings, if required.

Prerequisites

Before adding an application:

1. Configure the Enterprise Single Sign-On module.
2. Register a license in Management Console.
3. Prepare a template for the application.

Add an application

1. In the Management Console sidebar from the **Settings** area, select **Applications**.
2. Click **Add an application**.
3. In the **Add an application** window, select the Enterprise SSO integration module and the application template file, then click **Create**.

After the template is successfully uploaded, a page with application settings appears.

Adding to policy

To apply the application settings for users, add the application to a policy. To add an application to a policy, you must register license for this module.

To add an application to a policy:

1. Go to the **Policies** section.
2. Select a policy from the list.
3. In the policy card, go to the **Applications** tab.
4. Click **Add an application**.
5. From the drop-down list, select the application.
6. Click **Add**.

The application appears in the list of added applications.

Add an account

To set the user data that are applied when logging in to the ESSO application, add an account.

1. In the Management Console sidebar, open the **Policies** section.
2. Select a policy.
3. Go to the **Applications** tab.
4. To open the application integrated with the ESSO module, click the name of this application.
5. Click **Add an account**.
6. In the **Credentials type** field, select the type:
   - **Individual**: Each user has their own account to log in to the application.
   - **Active Director**: User data from **Active Directory** is used for login.
   - **Linked**: An account that is linked with one of the already existing accounts. Within one policy, you can synchronize accounts from different applications. This way you can have numerous applications with identical credentials. For linked accounts, you can only make changes in the account of the root application, rather than editing each entry separately.
   - **Shared**: All users have a single account to log in to the application.
7. In the **Description** field, enter a description of the account that is displayed in Management Console.
8. Configure the remaining settings depending on the selected credential type. If settings are already defined in the application template, these settings appear in Management Console automatically, but they cannot be edited.
9. Click **Save**.

The added account appears in the list of accounts for the application.

Set the authentication method

1. In the Management Console sidebar, open the **Policies** section.
2. Select a policy.
3. Go to the **Applications** tab.
4. To open the application integrated with the ESSO module, click the application name.
5. In the **Available authentication methods** section, select which methods can be used to log in to the integrated application.

6. Click **Save**.

Optional settings

Change general information

1. In the application card on the **General information** tab, click **Edit**.
2. Change the application name or description.
3. Click **Save**.

Upload a logo

1. In the application card on the **General information** tab in the **Logo** section, click **Upload**.

   The supported formats are JPG and PNG/ The maximum size is 512KB.

2. Select a file.
3. Click **Upload**.

Upload/download a template

1. In the application card on the **ESSO template** tab, click **Upload a template**.
2. In the **ESSO template** window, click **Select a file**.
3. Select a file.
4. Click **Upload template**.

To download the currently used template, click **Download a template**. The template becomes available in the **Downloads** folder.

Add an administrator

You can assign an administrator who will have access only to limited application settings. This user has the following rights:

- View all application settings in the **Applications** section.
- Add accounts for the application in the policy card.
- Add accounts for the application in the user profile.

To add an administrator:

1. Go to the **Administrators** tab.
2. In the **Object type** field, select the type:
    - **User**
    - **Group**
3. In the **Location** field, select the object location. It can be the entire user catalog or a separate container.
4. In the **Name** field, enter the full name or part of the name for search.
5. Click **Search**.
6. Select the object and click **Add**.

Create a shortcut for quick launch

For the application quick launch, you can create a shortcut that becomes available in the ESSO Agent context menu in the system tray.

To create a shortcut:

1. Go to the **Shortcuts** tab.
2. Click **Add a shortcut**.
3. In the window that appears, configure the following settings:
   - In the **Description** field, specify additional information for a shortcut that you create.
   - In the **Type** field, select **Command line**.
   - In the **Address** field, specify the full path to the executable file for the desktop application or the path to the browser executable file and the URL for web application.

Desktop

```
C:\TestFolder\Test.App.exe
```

Web

```
"%ProgramFiles%\\Internet Explorer\\iexplore.exe"
"https://full_dns/am/core/testapps/logon.html"
```

4. Click **Add**.

> **INFORMATION**
> If label parameters are specified when creating the template, these parameters automatically appear on the **Labels** tab.

Configure password policy

You can also configure a policy for password generation when changing it in the application.

To set up a password policy:

1. Go to the **Password policy** tab.
2. In the **Minimum password length** field, specify the minimum number of password characters.
3. In the **Maximum number of occurrences of each character** field, specify the number of occurrences of characters from the character group. To use this parameter, you need to create a character group.
4. Click **Add a group**.
5. In the **Group type** field, select the suggested character types or a custom group. When adding a custom group, specify your character group in the **Group characters** field.
6. In the **Number of characters** field, specify the number of characters.
7. Click **Add**. The added character group appears in the groups list.
8. Click **Save**.

Modify a client module

In the **Client modules** section, specify the target URL of the application or the path to the executable file. The value of this parameter is included in the application template. If required, this value can be changed.

To modify a client module:

1. Select the client module and click **Edit**.
2. Change the URL or path to the executable file.
3. If you need to use a regular expression, in the **Use regular expression** option, select **Yes**.
4. Click **Save**.

Configure user data caching

This setting allows enabling user data caching on the local computer for the Windows Logon component and applications integrated with the Enterprise Single Sign-On module.

If this setting is enabled for applications, you can log in to the system using an authenticator even without the physical network connection. If there is no connection, cached user data is used.

You can enable user data caching either in policy settings or individually for each user.

If a user is affected by a policy, you cannot change caching settings individually.

For more information about enabling this setting, see User data caching.

NPS Radius Extension

In the **Applications** section, you can add applications integrated with the NPS Radius Extension module, as well as configure additional settings, if required.

Prerequisites

Before adding an integrated application:

1. Configure the NPS Radius Extension module.
2. Register the license in Management Console.
3. Install and configure authentication providers.
4. Integrate the business application with NPS Radius Extension.

> ⚠ **IMPORTANT**
> If applications work through NPS (Network Policy Server) with installed RADIUS Extension, but authentication is not performed through Axidian Access, configure access in Management Console in the **Configuration** section. Otherwise, login into these applications is not available.

Add an application

To add an application integrated with NPS RADIUS Extension:

1. In the Management Console sidebar, open the **Applications** section.
2. Click **Add an application**.
3. In the opened window, select **NPS Radius Extension** and enter the application name. The name must be unique.
4. Click **Create**.

> ⓘ **NOTE**
> You cannot add another application if you have not changed the default IP address of the previously added application.

5. In the opened application window, go to the RADIUS tab.

6. Change the value of the **Application server IP address** field. You can specify multiple values separated by commas. The supported formats are IPv4 and IPv6. You can also use a mask or prefix to specify a subnet.

Address examples

Valid addresses

- 192.168.0.1/24
- 172.16.3.251
- 10.0.0.1
- 21:db8:58:74:1c60:2c18::4/64

Invalid addresses

- 192.168.0.300 — invalid IPv4;
- 172.16.3.251/33 — invalid network mask;
- fe80::1%eth0 — invalid IPv6;
- 21:db8:58:74:1c60:2c18:0:4/129 — invalid network mask;
- example.com — not an IP address.

Add to a policy

To apply application settings to users, add the application to a policy. You can only do this if you have a registered license for this module.

To add an application to a policy:

1. Go to the **Policies** section.
2. Select a policy from the list.
3. In the policy card, go to the **Applications** tab.
4. Click **Add an application**.
5. Select an application from the drop-down list.
6. Click **Add**.

The application appears in the list of added applications.

Configure an authenticator

For applications integrated with Radius Extension, you can select only one login method. When adding an application to a policy, the first available login method is enabled by default.

The list of available authentication methods displays providers installed on Core Server and supported by RADIUS applications.

Both one-factor and two-factor authentication are supported.

Prerequisites

Before configuring one-factor or two-factor authentication, install providers from the *Axidian AM <version number>\Axidian AM Providers\Axidian AM Radius Providers<version number>* directory on Core Server.

One-factor authentication

To configure one-factor authentication:

1. In the Management Console sidebar, open the **Policies** section.
2. Select a policy.
3. Go to the **Applications** tab
4. Select the application integrated with NPS Radius Extension.
5. In the **Authentication method** list, select a provider with the **1FA** prefix.
6. Click **Save**.

Available login methods for one-factor authentication include:

- Secured TOTP
- Hardware OTP
- Passcode
- Software TOTP
- Hardware TOTP

Two-factor authentication

To configure two-factor authentication:

1. In the Management Console sidebar, open the **Policies** section.
2. Select a policy.
3. Go to the **Applications** tab.
4. Select the application integrated with NPS Radius Extension.
5. In the **Authentication Method** list, select a provider with the **2FA** prefix.
6. Click **Save**.

Available login methods for two-factor authentication include:

- Passcode + Secured TOTP
- Passcode + SMS OTP
- Passcode + Software TOTP
- Passcode + Axidian Key (only in the push notification mode with login confirmation)
- Windows Password + Hardware HOTP
- Windows Password + Hardware TOTP
- Windows Password + Secured TOTP
- Windows Password + Software TOTP
- Windows Password + Storage SMS OTP
- Windows Password + Axidian Key (only in the push notification mode with login confirmation)
- Windows Password + Email OTP
- Windows Password + SMS OTP
- Windows Password + Telegram (only in the one-time password sending mode)
- One-string authenticator (authenticators are entered on a single line)

Optional settings

Change general information

1. In the application card, on the **General Information** tab, click **Edit**.
2. Change the application name or description.
3. Click **Save**.

Upload a logo

1. On the **General Information** tab in the **Logo** section, click **Upload**.

   The supported formats are JPG and PNG. The maximum image size is 512KB.

2. Select a file.

3. Click **Upload**.

Configure application access for users outside access policies

If required, you can enable or disable access to applications for users who are not included in the access policies, such as users from another domain.

1. In the application card, go to the **RADIUS** tab.
2. In the setting **If access parameters to the application are not set for a user**, enable or disable access.
3. Click **Save**.

Configure processing of duplicate authentication requests

If required, you can configure detection and processing of duplicate authentication requests. This option is useful if you have duplicate requests. For example, if a user does not have time to confirm login through a push notification and receives it again.

To configure processing of duplicate authentication requests:

1. In the application card on the **RADIUS** tab, enable the option **Enable detection and processing of duplicate authentication requests**.
2. In the **Request attributes for duplicate search** field, enter Radius attribute values.

   You can specify multiple values separated by commas. The default values are *1* (User-Name) and *264* (Request authenticator).

   Attributes can include username, password, IP address from which the authentication request is sent, and other parameters. The request is considered duplicate if the values of all specified attributes in the setting match the values in the cached request. For more information about supported RADIUS attributes, see the official Cisco documentation.
3. Select the response type for duplicate requests:
   - **Discard**: The RADIUS server does not send a response to a duplicate request.
   - **Reject**: If you have a duplicate request, the RADIUS server sends the *Access-Reject* response.
4. Click **Save**.

User data caching

This setting allows enabling user data caching on the local computer for the Windows Logon component and applications integrated with the Enterprise Single Sign-On module.

If this setting is enabled for applications, you can log in to the system using an authenticator even without the physical network connection. If there is no connection, cached user data is used.

You can enable user data caching either in policy settings or individually for each user.

If a user is affected by a policy, you cannot change caching settings individually.

For users within a policy

1. In the Management Console sidebar, open the **Policies** section.
2. From the list of policies, select the required policy.
3. Open the **Applications** tab and select the ESSO application or the Windows Logon component for which you need to enable data caching.

   > ⓘ **NOTE**
   > If you enable caching for one ESSO application, caching settings are applied to all ESSO applications within the policy.

4. Click **Data caching**.
5. For the **Allow caching account data on user's PC** setting, select *Yes*.
6. If required, configure the cache lifetime:

   - In the **Initial date** and **Final date** fields, set the expiration date for cached authenticators.
   - In the **Number of days since last entry to online mode** field, set the validity period of cached authenticators, counting the period from the last login to the system with network connection.

   If the validity period of cached authenticators has expired and Core Server is unavailable, the following error message appears during login: *Login error. The user data caching period*

has expired.

7. Click **Save**.

For an individual user

1. In the Management Console sidebar, open the **Users** section.
2. Find the user for whom you need to enable caching.
3. Open the **Applications** tab and select the ESSO application or the Windows Logon component for which you need to enable data caching.

> ⓘ **NOTE**
> If you enable caching for one ESSO application, caching settings are applied to all ESSO applications of the user.

4. Click **Data caching**.
5. For the **Allow caching account data on user's PC** setting, select *Yes*.
6. If required, configure the cache lifetime:
   - In the **Initial date** and **Final date** fields, set the expiration date for cached authenticators.
   - In the **Number of days since last entry to online mode** field, set the validity period of cached authenticators, counting the period from the last login to the system with network connection.

   If the validity period of cached authenticators has expired and Core Server is unavailable, the following error message appears during login: *Login error. The user data caching period has expired*.
7. Click **Save**.

For more information about login settings when network is unavailable, see the Enterprise SSO and Windows Logon sections.

Licenses

Axidian Access uses two types of licenses:

- Basic Axidian Access license
- Licenses for additional modules

For more information about licensing, see Licensing.

Obtain

Licenses are generated based on the installation identifier. To locate the identifier and obtain licenses:

1. In the Management Console sidebar, in the **Settings** section, select **Licenses**.
2. On the **License files** tab, copy the value from the **Installation identifier** field.
3. Send the copied value to the manager assigned to your organization or to technical support at https://support.axidian.com/support/home.

Register

After you obtain a license, register it by performing the following steps:

1. In the Management Console sidebar, in the **Settings** section, select **Licenses**.
2. On the **License files** tab, click **Upload a license**.
3. Click **Select a file**.
4. Select the required license file and click **Register**.

After successful license registration, the licenses are displayed in the table.

Issue a license to a user

For all users who are included in access policies, licenses are issued automatically. The license is assigned to the user after successful authentication in the corresponding module.

Terminate a license

To terminate a license, an automatic license release mechanism is used. This mechanism is enabled by default.

A license is revoked from a user in the following cases:

- The user is removed from a policy.
- The user is disabled or moved out of the policy scope.
- The user is blocked or deleted from the user catalog.

If the license revocation mechanism is disabled, licenses are not released even when these conditions are met.

In Axidian Access, you can also configure the time after which the license is terminated and restart the revocation mechanism if licenses are not released automatically. To do this, edit the server configuration file *C:\inetpub\wwwroot\am\core\Web.config*. For more information, see Configuring the license revocation mechanism.

Distribute by policies

To limit the number of licenses used for a department (container), you can set the number of available licenses for a specific policy.

To set the number of available licenses for a policy:

1. In the Management Console sidebar, go to the **Policies** section.
2. Select a policy from the list.
3. On the policy page, go to the **Licenses** tab.
4. Select licenses and click **Available quantity**.
5. In the window that appears, enter the number of available licenses for a policy and click **Save**.

In the **License distribution by policies** section, the number of licenses used within each policy is displayed.

On the policy page, you can view the number of licenses used in this policy and the total number of available licenses for this policy.

Update

1. Contact the manager assigned to your organization or technical support at https://support.axidian.com/support/home with a request to create new licenses.
2. Register new licenses in Management Console.

After the licenses expire, the new licenses take effect within the specified date and time.

You can track license expiration dates in Management Console in the **Licenses** section.

Authenticators

Depending on your scenario, you can configure authenticators in the following locations of Axidian Access:

- **Configuration→Authenticators**
- **Applications** tab in the policy card
- **Authenticators** tab in the user profile

General settings

The **Configuration→Authenticators** section contains general settings for all authenticators installed on the Axidian Access server. These settings apply to all users. Settings vary depending on the selected authenticator.

Configure automatic block

For most authenticators, you can configure automatic blocking in case of unsuccessful login attempts.

> ⓘ **NOTE**
> You cannot configure authenticator blocking in Management Console for two-factor authentication within RADIUS Extension applications.

To configure automatic blocking in Management Console, perform the following actions:

1. Select an authenticator.
2. For **Block authentication method after several unsuccessful attempts**, set the value to *Yes*.
3. Set values for the following parameters:
    - **Number of authentication attempts before blocking**. This parameter determines the number of unsuccessful authentication attempts before the login method is blocked. The login method remains unavailable until the administrator unblocks it or until the blocking timeout expires. The default value is *5*. The specified value must be greater than or equal to *1*.

- **Reset blocking counter after ... minutes**. This parameter determines how many minutes must pass after an unsuccessful login attempt and before the counter of unsuccessful attempts is reset to *0*. The default value is *5*. The specified value must be greater than or equal to *1*. Time interval before the blocking counter is reset must not exceed the login method blocking timeout, unless the latter value equals *0*.

- **Login method blocking timeout (in minutes)**. This parameter determines the period during which the specified login method is blocked. After the timeout expires, the login method is automatically unblocked. The default value is *5*. If set to *0*, the login method remains unavailable to a user until the administrator unblocks it in Management Console.

4. Click **Save**.

Configure mandatory verification

You can configure mandatory verification to register Software TOTP, Secured TOTP, and Software HOTP authenticators.

If you enable authenticator mandatory verification, an additional confirmation window appears during the registration of authenticators.

Prerequisites

This setting is available in Management Console for version 8.2.6 or higher. If an older version is installed, an update is required. You can view the version information in the **Help and Support** section of the Management Console sidebar.

To add the mandatory verification setting to Management Console, use the *EA.Settings.Migration.Tool* utility.

To enable mandatory verification:

1. In the **Configuration→Authenticators** section, select Software TOTP, Secured TOTP, or Software HOTP.
2. In the **Main settings** section, enable mandatory verification.

To confirm registration, enter the received authentication data and click **Confirm**.

> ⓘ **NOTE**
> When registering Storage SMS, mandatory verification is performed by default.

Policy settings

You can also configure authenticator settings that apply to all users within the policy. These settings are available in the policy card in Management Console.

To configure authenticator settings for users within the policy, perform the following actions:

1. In the Management Console sidebar, select **Policies**.
2. Select a policy.
3. Go to the **Applications** tab. This tab contains the information about applications and integration modules added to the policy. For more information about policy settings, see Policies.
4. Select an application.
5. In the **Available authentication methods** section, enable or disable using authenticators.
6. For Identity Provider, Windows Logon, ESSO Agent, and ADFS Extension, select the Axidian Key Provider operating mode.
7. For Identity Provider, ADFS Extension, and IIS Extension, select the Telegram Provider operating mode.

Settings for an individual user

You can configure authenticator settings for an individual user in user profiles of Management Console.

To open a user profile:

1. In the Management Console sidebar, select **Users**.
2. Locate the user. For more information about search settings, see Users.
3. In the user profile, go to the **Authenticators** tab. This tab contains the information about the number of registered user authenticators and their parameters.

Register

Before registering authenticators, create a policy and apply it to end users.

> ⓘ **NOTE**
> Email OTP and SMS OTP providers do not require registration and can be used if a user has specified an email or a phone number. If parameters are not specified, the authenticators are not displayed in the list of available authenticators for a user.

To register an authenticator:

1. Click **Register** and from the drop-down list, select an authentication method.

   If a user has registered the maximum number of authenticators for a specific method, then this authentication method is be displayed.

2. Configure additional settings to complete registering an authenticator. The windows view and available actions vary depending on the selected authenticator.

3. After successful registration, the authenticator status is *Active*.

4. If a provider with registered authenticators was deleted, then the authenticator status is *Not Installed*.

The authenticator appears in the list of registered authenticators for a user.

Disable

You can restrict using an authenticator.

> ⓘ **NOTE**
> You cannot disable the Windows Password authentication method. We recommend that you deactivate a user account in Active Directory if you need to block the domain password.

To disable using an authenticator:

1. Select an authenticator.
2. Click **Disable**.

After disabling the authenticator, a user cannot use this login method. The authenticator status is *Disabled*.

In the user profile, you can disable using an authenticator for a single user. You can disable using an authenticator for all users in the Configuration→Authenticators section.

Enable

To enable using an authenticator:

1. Select an authenticator.
2. Click **Enable**.

After you enable the authenticator, a user can start using this login method. The authenticator status is *Active*.

> **NOTE**
> In the user profile, you can enable using an authenticator only if this authenticator is disabled for an individual user. You can enable using an authenticator for all users only in the Configuration→Authenticators section.

Unblock

If automatic unblocking after unsuccessful login attempts is disabled for an authenticator, you need to unblock it manually in the user profile. To perform this action:

1. Select the blocked authenticator.
2. Click **Unblock**.

The login method becomes available to a user.

Delete

1. Select an authenticator.
2. Click **Delete**.
3. Confirm deletion.

The authenticator disappears from the list of authenticators registered for a user.

Users

The user profile includes the information about users from Active Directory (AD) and user settings in the Axidian system.

Locate a user

1. In the Management Console sidebar, open the **Users** section.
2. In the **Name** field, enter the username, part of the name, or UserPrincipalName (UPN, username in email address format, such as *username@domain.com*).
3. In the **Container** field, select the search location.

   It can be the entire container or a specific organizational unit. If you leave the **Name** field empty, the search results will display all users in the selected container.
4. Click **Search**.

The window displays the list of users that match the search criteria.

View general information

1. Locate a user.
2. To open the user profile, click the name of the user.
3. Click the **General information** tab to view the following information:
   - From Active Directory:
     - **Photo**: The `jpegPhoto` or `thumbnailPhoto` attributes. If the attributes are empty, a placeholder is used.
     - **Account**: The `userPrincipalName` attribute.
     - **Email**: The `mail` attribute.
     - **Phone**: The `telephoneNumber` attribute.
     - **Path**: The location in Active Directory.
   - From Axidian Access:
     - **Policy**: The name of the policy that applies to a user. To go to the policy card, click the policy name.
     - **Applications count**: The number of applications available to a user.

- **Authenticators count**: The total number of authenticators registered for a user.
- **Events**: Five most recent events related to a user. To view all events related to a user, click **All user events**.

Adding users to a policy

To apply policy settings to a user, you need to add this user to the policy scope. For more information, see the Policies section.

View application information

If a user is within the scope of more than one policy, they have access to all applications and modules added to these policies. To learn more about how to manage policies in Management Console, see the Policies section.

To view the list of applications and modules available to a user:

1. In the user profile, go to the **Applications** tab.
2. The opened window displays a table with a list of applications and modules available to a user.
3. To view more detailed information about authentication methods configured for a module or application, click its name.

Configure data caching

You can enable data caching for a user on the local computer for the Windows Logon component and applications integrated with the Enterprise Single Sign-On module. Data caching allows logging in to the system using the authenticator even without physical network connection. For more information about enabling this setting, see the User data caching section.

Configure authenticators

In the user profile, you can manage authenticator settings for a user. For more information, see Authenticators.

View login history

In the user profile, you can view the history of user logins to Axidian Access modules. Also, you can export this information to a file in the CSV, XLSX, or PDF formats.

To view the user login history:

1. In the user profile, go to the **Sign in history** tab.
2. If required, set the following search filters:
    - Specify the start and end dates of the period for which you want to view the login history.
    - Select an application or a module.
    - Select the type of authenticator used for login.
    - Select the event state: **Successful sign in**, **Failed to sign in**, or **Not set**.
3. Click **Apply**.
4. To export the login history, click **Export** and select the file format. The supported formats are CSV, XLSX, and PDF.

View licenses

For all users who are included in access policies, licenses are registered automatically. The license is assigned to a user after successful authentication in the corresponding module.

To view the licenses available to a user, go to the **Licenses** tab in the user profile.

Policies

Policies allow you to configure access settings for a specific group of users.

Create a policy

1. In the Management Console sidebar, open the **Policies** section.
2. Click **Create a policy**.
3. Specify values for the following parameters:
   - **Priority**. The higher the value, the higher the priority.
   - **Name**
   - **Description** (optional)

Configure a policy

In the policy card, you can configure policy parameters.

1. In the Management Console sidebar, open the **Policies** section.
2. To open the policy, click the policy name.
3. Go to the required tab and perform the corresponding configuration.

Edit information

In the policy card, you can change the previously set name, priority, and description of the policy.

1. On the **Information** tab, click **Edit**.
2. Make the required changes.
3. Click **Save**.

Add/remove an application

If you need to configure a policy, add applications that are affected by this policy:

1. In the policy card, go to the **Applications** tab.
2. Click **Add an application**.

3. In the **Add an application** window, select the required application.
4. Click **Add**.

   The application appears in the table with all added applications.

> **NOTE**
> You can only add an application if you have a registered license for this module.

To remove an application:

1. In the policy card, go to the **Applications** tab.
2. Select an application.
3. Click **Remove** and confirm the removal.

   The application will be removed from the list of added applications.

For more information about application settings, see the Applications section.

Add/remove objects in scope

To apply policy settings to specific users, add objects to the scope. You can add individual users, groups, or departments.

To add an object to the scope:

1. In the policy card, go to the **Scope** tab and click **Add**.
2. In the **Object type** field, select the type:
   - **User**
   - **Group**
   - **Department**
3. In the **Location** field, select the object location. It can be the entire user catalog or a separate container.
4. In the **Name** field, specify the full or partial name of the object and click **Search**.

If you leave the **Name** field empty, the search results display all objects of the specified type that are in the selected location.

5. Select an object.
6. Click **Add**.

The object appears in the list of scope objects.

A global administrator can also search for and remove objects from the policy scope.

You can use templates defined in the Management Console configuration file to search for objects.

To remove an object from the scope:

1. In the policy card, go to the **Scope** tab.
2. Select an object.
3. Click **Delete** and confirm the deletion.

   The object will be removed from the list of objects in the scope.

The global administrator can also use the object search function.

To locate and delete an object in the policy scope:

1. In the policy card, go to the **Scope** tab.
2. Click **Search**. The search page appears.
3. In the **Object type** field, select the type:
   - **User**
   - **Group**
   - **Department**
4. In the **Location** field, select the object location. It can be the entire user catalog or a separate container.
5. In the **Name** field, specify the full or partial name of the object and click **Search**.

   If you leave the **Name** field empty, the search results display all objects of the specified type that are in the selected location.

6. Select an object from the search results, click **Delete** and confirm the deletion.

The object will be removed from the list of objects in the scope.

Assign roles

You can assign users who can have the role of administrator, operator, or inspector for this policy. You can assign a role to individual users or a group of users. Each role has its own set of rights.

To assign a role:

1. In the policy card, go to the **Administrators** tab and click **Add**.
2. Select a role:
   - **Administrator**
   - **Operator**
   - **Inspector**
3. In the **Object type** field, select the required type:
   - **User**
   - **Group**
4. In the **Location** field, select the object location. It can be the entire user catalog or a separate container.
5. In the **Name** field, specify a full or partial name of the object and click **Search**.

   If you leave the **Name** field empty, the search results display all objects of the specified type that are in the selected location.
6. Select an object.
7. Click **Add**.

After you add an object, the object is displayed in the list on the **Administrators** tab.

Distribute licenses

You can set the available number of licenses for a specific policy to limit the number of licenses used for a container. For more information on how to do this, see the Licenses section.

Delete a policy

1. In the Management Console sidebar, open the **Policies** section.
2. In the policy list, select the required policy.
3. Click **Delete** and confirm the deletion.

The deleted policy will disappear from the list of policies.

Also, you can delete a policy from the policy card by performing the following steps:

1. In the policy card, go to the **Information** tab.
2. Click **Delete** and confirm the deletion.

User data caching

This setting allows enabling user data caching on the local computer for the Windows Logon component and applications integrated with the Enterprise Single Sign-On module.

If this setting is enabled for applications, you can log in to the system using an authenticator even without the physical network connection. If there is no connection, cached user data is used.

You can enable user data caching either in policy settings or individually for each user.

If a user is affected by a policy, you cannot change caching settings individually.

For more information on how to enable this setting, see User data caching.

Devices

In the **Devices** section, you can manage general settings for Software HOTP, Hardware OTP, and TOTP devices. To learn more about particular device settings, see the Authentication provider installation and configuration section.

Add a device

1. In the Management Console sidebar, open the **Devices** section.
2. Click **Add from a file** to upload the information about device parameters or click **Add** to set device parameters manually.

Add from a file

1. Click **Add from a file**.
2. Select a file with device parameters and click **Add**.
3. The **Result** window shows the number of added devices.
4. Click **Close**.

The added device appears in the list of available devices.

Add manually

1. Click **Add**.
2. In the **Serial number** field, specify the device serial number.
3. In the **Secret key** field, specify the device key (parameter `Seed`).
4. If required, specify additional information in the **Comment** field.
5. Click **Add**.

The added device appears in the list of available devices.

Delete a device

1. In the Management Console sidebar, open the **Devices** section.

2. To delete a device manually, select it from the list and click **Delete**.

   If required, you can perform a search using filters at the top of the page.

3. To delete a device using the information from a file:

   1. Click **Delete from a file**.
   2. Select the XML file with device parameters and click **Delete**.
   3. The **Result** window shows the number of deleted devices.
   4. Click **Close**.

Synchronize a device

To synchronize a device if the counter on a device and the server is desynchronized:

1. In the Management Console sidebar, open the **Devices** section.
2. Click the device serial number.
3. On the top panel, click **Synchronize OTP**.
4. In the opened window, enter two one-time passwords consecutively generated on the device and click **Synchronize**.

Block/unblock a device

In Management Console, you can disable using the added device. Also, you can unblock a previously blocked device.

To block a device:

1. In the **Devices** section of Management Console, select a device from the list of added devices.
2. Click **Block**.

The device status changes from **Active** to **Blocked**.

To unblock a device:

1. In the **Devices** section of Management Console, select a blocked device.
2. Click **Unblock**.

The device status changes from **Blocked** to **Active**.

Events

All Axidian Access event records are stored in the event log on each Axidian Access server. Axidian Access keeps records of the following types:

- Error
- Information
- Warning

To open the event log, from the left menu in the **Audit** section, select **Events**.

For more targeted event search, use the following filter parameters:

- *User or initiator*: Select the account where the event is performed.
- *Application*: Filter events by the selected Axidian Access component.
- *Start/End date*: Specify the time interval within which events are selected.
- *Authenticator type*: Filter events by the selected authentication method.
- *Module*: Filter events by the selected Axidian Access system module.

  > ⓘ **NOTE**
  > When searching for events, events for a specific system module are displayed. As for the *Application* parameter, events are displayed for a specific system application with numerous modules available.

- *Description contains*: Specify the full value of the event attribute, such as application, administrator, or computer.

  > ⓘ **INFORMATION**
  > Partial search using the asterisk symbol (*) is not possible because of the limitations of Microsoft EventLog. If you use other supported storage methods, the search capabilities are limited by the capabilities of EventLog itself.

- *Event type*: Specify the event type, such as error, information, and warning. If you select the **Not specified** option, the search result shows events of all types.

- *Event code*: Select a specific Axidian Access system event.

To export events, click **Export** at the top of the page. The supported export formats include CSV, XLSX, and PDF.

Configuration

In this section of Management Console, you can configure authenticators and set access rights.

Access rights

Roles

In the **Roles** section, you can view the rights that roles have across the entire Axidian Access system — global administrator, operator, or inspector. Each role has its own set of rights. For more information about roles, see Roles.

Global administrators

In the **Global administrators** section, you can view and add users with the global administrator, operator, or inspector roles, which grant rights to administer the entire Axidian Access system.

To assign a role:

1. Go to the **Global administrators** tab.
2. Click **Add** and select the required role.
3. In the **Name** field, enter the username, part of the name, or UPN.
4. In the **Object type** field, select the type — **User** or **Group**.
5. In the **Location** field, select the object location. It can be the entire user catalog or a separate container. If you leave the **Name** field empty, the search results will display all objects of the specified type in the selected location.
6. Select an object.
7. Click **Add**.

The user with the assigned role appears in the list on the **Global administrators** tab.

You can also assign a user a role within a single policy.

Authenticators

In this section, you can view and configure the authenticators installed on this server.

Enable/Disable using and editing

To configure using and editing authenticators:

1. Select an authenticator.
2. In the **Main settings** section:
    - In the **Disable using** setting, specify whether users can use or register the selected authenticator.
    - Configure other settings depending on the authenticator type.
3. For authenticators that require registration, in the **Actions available to users** section:
    - In the **Registration of new authenticators** setting, specify whether users can register new authenticators.
    - In the **Editing existing authenticators** setting, specify whether users can modify already registered authenticators.
    - In the **Deleting of existing authenticators** setting, specify whether users can delete already registered authenticators.
    - In the **Enable editing authenticator comment** setting, specify when users can edit comments for already registered authenticators.

Change the maximum number

In this section, you can define the maximum number of authenticators that a single user can register. This setting is available for the following authenticators:

- IronLogic Z2USB Provider
- Passcode Provider
- OMNIKEY Provider
- Smart Card Provider

To set the maximum number:

1. Select an authenticator.
2. In the **Maximum number** field, enter a value.
3. Click **Save**.

User help

User profile

User information

Manage authenticators

Managing user authenticators

Localization

Change language in User Console

Manage Windows Password

Change or register domain password

User profile

The user profile contains the following information from the Active Directory profile:

- **Account name**: The `userPrincipalName` attribute.
- **Path**: The location in Active Directory.
- **Email**: The `mail` attribute.
- **Phone**: The `telephoneNumber` attribute.
- **Photo**: The `jpegPhoto` or `thumbnailPhoto` attributes. If the attributes are empty, a placeholder is used.
- **Login history**: Recent events in Axidian Access initiated by a user.
- Information about registered authenticators on the **Authenticators** tab.

Manage authenticators

In the user profile **Authenticators** tab, you can view the information about the number of registered authenticators and their parameters. By default, users can only register authenticators in User Console. For other actions with authenticators, request permissions from the administrator.

Register an authenticator

> **NOTE**
> Email OTP and SMS OTP providers are registered automatically if a user has an email or phone number specified in Active Directory.

To register an authenticator:

1. Select an authenticator.
2. Click  .
3. Select **Register**.
4. Enter the requested information and click **Save**.

   The requested information and the registration procedure in general vary depending on the authenticator.

5. If required, enter the authentication data to confirm registration.

After successful registration, the authenticator is displayed as registered.

If a user has registered the maximum number of authenticators, registration fails with the error *"Access denied: Maximum number of authenticators reached."*

If the administrator deletes a registered authenticator, it is no longer displayed in the user profile.

Edit an authenticator

1. Select an authenticator.
2. Click  .
3. Select **Edit**.

4. Enter new data for the authenticator.
5. Click **Save**. If required, enter the authentication data to confirm registration.

Delete an authenticator

1. Select an authenticator.
2. Click  .
3. Select **Delete** and confirm the deletion.

Localization

To change the language:

1. In the upper right corner, click the language label.
2. From the drop-down list, select the required language.

Manage Windows Password

In User Console, you can register or change your domain password.

Register a domain password

1. Open User Console.
2. Go to the **Authenticators** tab.
3. In the authenticators list, select **Windows Password**.
4. Click .
5. Select **Register a password**.
6. In the opened window, enter a password and click **Save**.

Change a domain password

1. Open User Console.
2. Go to the **Authenticators** tab.
3. In the authenticators list, select **Windows Password**.
4. Click .
5. Select **Change a password**.
6. In the opened window, enter your current password, new password, confirm your new password, and click **Save**.

API

General information

General information

Quick start

Quick start

API methods

8 items

Use cases

6 items

General information

You can use the Axidian Access API to integrate with applications.

This section describes the main points and common parameters that are required for using the Axidian API.

API request

To access a method, run a `POST` request or, for some methods, a `GET` request to *http(s)://<dns_axidian_server>/am/core/<url_api_method>*, where:

- *dns_axidian_server* — the full DNS name of the server where Axidian Access Server is deployed.
- *url_api_method* — the address of the corresponding method. The addresses are specified in Method Documentation or in Swagger.

Example for the authenticateByWindowsToken method, where the DNS name of the Axidian Access server is *axidian.access.local*:

```
https://axidian.access.local/am/core/api/v5/logon/authenticateByWindowsToken
```

> ⚠ **IMPORTANT**
> When using the HTTPS protocol, upload a valid certificate to the Axidian Access server.

`UserId`

This parameter specifies the unique identifier of the user from the Axidian system.

The identifier consists of a string ID specified in *web.config* in the `userId` parameter and the user's `objectGuid` from Active Directory.

For example, the `userId` parameter specifies the user's GUID value *10efa04f-7ba9-47d8-89db-56e166f1679f*. UserId is `UserId\_10efa04f-7ba9-47d8-89db-56e166f1679f`.

To get the userId, use the `[POST]/api/v5/user/searchUserId` method.

`modeId`

The unique identifier of the Axidian authentication method.

Authentication methods:

- Windows Password Provider {CF189AF5-01C5-469D-A859-A8F2F41ED153}
- Smart Card Provider {0AF65AD8-DB77-4B64-B489-958D9B36E28C}
  - +PIN {42456525-8EC1-4AB1-97B8-45AF8635D10F}
- IronLogic Z2USB Provider {CB5109DA-B575-422C-8805-524FE12B02F5}
  - +PIN {1EDFD6E1-FD66-4A5B-971A-CBA0C611BA9B}
- OMNIKEY Provider {4B15AF52-A795-4CA6-B7CD-CDB8ABF2D2C2}
  - +PIN {199B8FA8-FA9D-4ED8-941D-F86A5C681E0C}
- Futronic Provider {A0EF00AD-1EEB-4D48-8BCF-06E19CD5585F}
- Passcode Provider {F696F05D-5466-42b4-BF52-21BEE1CB9529}
- Email OTP Provider {093F612B-727E-44E7-9C95-095F07CBB94B}
- Hardware TOTP Provider {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
- HOTP Provider {CEB3FEAF-86ED-4A5A-BD3F-6A7B6E60CA05}
- Storage SMS OTP Provider {3F2C1156-B5AF-4643-BFCB-9816012F3F34}
- Telegram Provider {CA4645CC-5896-485E-A6CA-011FCC20DF1D}
- SMS OTP Provider {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}
- Software OTP Provider {0FA7FDB4-3652-4B55-B0C0-469A1E9D31F0}
- Axidian Key Provider {DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68}
- MFA Provider {070719BA-EB57-4EA8-BB4D-D15A33E7363D}

To install all the authentication methods on the Axidian Access server, use the `[GET]/api/v5/authenticationProvider/getAll` method.

`ApplicationId`

Internal module

> ⚠ **IMPORTANT**
> You do not need a license to work with an internal module, it is enough to have rights to the operations provided by the internal modules.

The functionality of Axidian Access is divided into groups that are handled by internal modules. They implement the internal business logic of the solution. Their string identifiers are used to access different parts of the Core Server API.

Internal modules of Core Server:

- User Profile Setting Management
- User Access Control Management
- Hardware Devices Management
- License Management
- Authenticator Management
- Authenticator Enrollment
- User Accounts Management
- Policy Management
- User Cache Management
- Business Applications Management

In addition, there is also a set of internal applications:

- Enterprise Management Console
- Self Service
- Native Enroller

Integration module

> ⚠ **IMPORTANT**
> To work with integration modules, a license for the corresponding module is required.

Integration module is a module that implements the interconnection of Axidian Access and external systems with which the integration is performed. These include:

- Windows Logon
- Enterprise SSO
- IIS Extension
- NPS RADIUS Extension
- Identity Provider
- ADFS Extension

- RDP Windows Logon
- Authentication API

Integration modules can be a single business application, such as Windows Logon, or can have a set of business applications that can be added or removed, such as the Enterprise SSO module. It provides the ability to integrate with different types of user applications (desktop or web applications), and for each one you need to create a separate ESSO business application: 1C, Lotus Notes, and so on.

Policy identifier

To locate the policy identifier, do the following:

1. Open Management Console.
2. Go to the **Policies** tab.
3. Select the target policy.
4. Copy the policy identifier from the `policyId` parameter in the URL.

Quick start

To call API methods and test requests, use the Swagger interface, which is built into Core Server.

To enable access to the interface, do the following:

1. Open the Core Server configuration file *web.config* located in *C:\inetpub\wwwroot\am\core*.
2. Set the `enableSwagger` parameter to *true*.

The Swagger interface will be available at the link
http(s)://<DNS_Axidian_Server>/am/core/swagger/

Request administrator token

To perform most API requests, certain rights in the Axidian system are required. To successfully perform such requests, you need to authenticate in the API as a user with the required set of rights. After you complete the authentication, you get a session token that is used in API requests.

Get a token using `authenticate ByWindowsToken`

Method description

This is the simplest method of getting a token. To get a token, call a single method on behalf of the user for whom the token will be issued:

- If you perform the request via a browser, then you must start the browser on behalf of the required user.
- In PowerShell, when making the request you need to use the additional parameter `UseDefaultCredentials` (example of the `postRequestForWin` function in PowerShell).

Any application from the Axidian list can be specified as the application identifier.

> ⚠ **IMPORTANT**
> The application used is specified in the authentication events.

Response Body

```
{
  "ValidPropertiesMask": 5,
  "Token": ""
  "LogonResult": null,
  "UserId": ""
}
```

Get a token using `authenticate`

Method description

In this method of getting a token, you can use the authentication methods that are available to the user.

As a result of authentication, a similar token is obtained.

API methods

Authenticator

getUserAuth, findByUserIds

License

getAllLicenses, getCatalogObjectLicenses

Logon

isAvailable, authenticate, getAvailableMethods, authenticateByWindowsToken

Policy

get

TemplateSession

createTemplate, openEnrollSession, openVerifySession, openIdentifySession, prepareTemplateData

User

searchUserId

UserCatalog

getObjects, searchUsers, searchGroups, searchContainers


UserProfile
findLogonInfoByUser

Authenticator

`getUserAuth`

Returns all of the user's enrolled authenticators by the internal identifier in Axidian Access.

```
POST /api/v5/authenticator/getUserAuth
```

Request object

```
{
  "Id": "00000000-0000-0000-0000-000000000000",
  "UserId": "string",
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `Id` — required parameter. Internal identifier in Axidian Access.
- `UserId` — required parameter. Internal identifier of the user in Axidian Access.
- `AccessToken` — required parameter. Token of the administrator on whose behalf the API request is performed.
- `ApplicationId` — required parameter. String identifier of the Axidian Access module.

Response object

```
{
  "AuthenticationCountLimit": 0,
  "CreatedOn": "2022-10-26T07:23:08.550Z",
  "Description": "string",
  "ExpirationDate": "2022-10-26T07:23:08.550Z",
  "Flags": 1,
  "Id": "00000000-0000-0000-0000-000000000000",
  "IsDisabled": true,
  "IsLocked": true,
  "ModeDeviceName": "string",
  "ModeId": "00000000-0000-0000-0000-000000000000",
  "ModeType": 0,
  "ModeTypeName": "string",
  "UserId": "string",
  "SerialNumber": "string",
  "Device": {
    "Id": "00000000-0000-0000-0000-000000000000",
    "AuthType": "00000000-0000-0000-0000-000000000000",
    "SerialNumber": "string",
    "RegistrationDate": "2022-10-26T07:23:08.550Z",
    "IsEnabled": true,
    "LastUserId": "string",
    "Comment": "string",
    "Model": "string"
  },
  "CreatedBy": "string"
}
```


`findByUserIds`

Returns the internal identifier of the user in Axidian Access.

```
POST /api/v5/authenticator/findByUserIds
```

Request object


```
{
  "UserId": "string",
  "AccessToken": "string",
  "ApplicationId": "string"
}
```


In the request object:

- `UserId` — required parameter. Internal identifier of the user in Axidian Access.

- `AccessToken` — required parameter. Token of the administrator on whose behalf the API request is performed.
- `ApplicationId` — required parameter. String identifier of the Axidian Access module.

Response object

```
[
  "00000000-0000-0000-0000-000000000000"
]
```

License

`getAllLicenses`

Returns all licenses registered in Axidian Access.

```
POST /api/v5/license/getAllLicenses
```

Request object

```
{
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `AccessToken` — required parameter. Token of the administrator on whose behalf the API request is performed.
- `ApplicationId` — required parameter. String identifier of the Axidian Access module.

Response object

```
{
  "LicenseUid": "string",
  "InstanceId": "string",
  "Type": "string",
  "Amount": 0,
  "BeginDate": "2022-10-26T08:43:22.406Z",
  "EndDate": "2022-10-26T08:43:22.406Z",
  "Description": "string",
  "Issuer": "string",
  "IssuedTo": "string",
  "IssueDate": "2022-10-26T08:43:22.406Z",
  "Id": "00000000-0000-0000-0000-000000000000",
  "ControlValue": "string"
}
```

`getCatalogObjectLicenses`

Returns the registered licenses for a specific catalog object.

```
POST /api/v5/license/getCatalogObjectLicenses
```

Request object

```
{
  "ObjectId": "string",
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `ObjectId` — identifier of the catalog for which the registered licenses will be retrieved.
- `AccessToken` — required parameter. Token of the administrator on whose behalf the API request is performed.
- `ApplicationId` — required parameter. String identifier of the Axidian Access module.

Response object

```
{
  "ApplicationId": "string",
  "CatalogObjectId": "string",
  "PolicyId": "00000000-0000-0000-0000-000000000000"
}
```

`getAcquiredLicenseCount`

Returns the number of used licenses of the specified type in the selected policy.

```
POST /api/v6/license/getAcquiredLicenseCount
```

Request object

```
{
  "LicenseType": "string",
  "PolicyId": "00000000-0000-0000-0000-000000000000",
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `LicenseType` — type of license.
- `PolicyId` — policy identifier. The identifier can be obtained from the policy URL in Management Console or using the `/api/v6/policy/getAll` method.
- `AccessToken` — required parameter. Token of the administrator on whose behalf the API request is performed.
- `ApplicationId` — required parameter. String identifier of the Axidian Access module.

Response object

```
"0"
```

Logon

`isAvailable`

Checks the availability of user authentication in the application.

- The `ApplicationId` parameter specifies the identifier of the application.
- The `UserId` parameter specifies the identifier of the target user.
- The `ExcludeWindowsPassword` parameter specifies *true\false*:
  - If *true* is specified, then authentication by the domain password is excluded from the check. In that case any other authentication method must be available to the user, otherwise the request returns *false*.
  - If *false* is specified, the domain password is not excluded from the possible authentication methods in the scenario.

```
[POST] /api/v5/logon/isAvailable
```

Request object

```
{
  "ApplicationId": "string",
  "UserId": "string",
  "ExcludeWindowsPassword": true
}
```

in the request object:

- `ApplicationId` — required parameter. String identifier of the Axidian Access module.
- `UserId` — required parameter. Identifier of the user in Axidian Access.
- `ExcludeWindowsPassword` — optional parameter. Default value *true*.

Response object

```
true\false
```

`authenticate`

Performs user authentication in the *BusinessApplication* application using the registered template *TemplateId*.

```
[POST] /api/v5/logon/authenticate
```

Request object

```
{
  "TemplateId": "00000000-0000-0000-0000-000000000000",
  "BusinessApplication": "string"
}
```

Response object

```
{
  "Token": "string",
  "LogonResult": {
    "Result": "string"
  },
  "UserId": "string"
}
```

Example of a successful response object

```
{
  "ValidPropertiesMask": 5,
  "Token": "eyJ0e----.eyJleH----",
  "LogonResult": {
    "ValidPropertiesMask": 0,
    "Result": null
  },
  "UserId": "UserId_b1cfaa29-6368-4c50-9868-06dbbe21fe23"
}
```

`getAvailableMethods`

Returns an array of available authentication methods for the user *UserId* in the application specified in *ApplicationId*.

```
[POST] /api/v5/logon/getAvailableMethods
```

Request object

```
{
  "ApplicationId": "string",
  "UserId": "string",
  "IncludeModeIds": [
    "00000000-0000-0000-0000-000000000000"
  ],
  "ExcludeModeIds": [
    "00000000-0000-0000-0000-000000000000"
  ]
}
```

in the request object:

- `ApplicationId` — required parameter. String identifier of the Axidian Access module.
- `UserId` — required parameter. Identifier of the user in Axidian Access.
- `IncludeModeIds` — optional parameter. GUIDs of the Axidian authentication methods that will be taken into account when checking the possibility of authentication.
- `ExcludeModeIds` — optional parameter. GUIDs of the Axidian authentication methods that will be excluded when checking the possibility of authentication.

Response object

```
{
  "AutheticationMethods": [
    "00000000-0000-0000-0000-000000000000"
  ]
}
```

Response examples

No available methods

```
{
  "ValidPropertiesMask": 1,
  "AutheticationMethods": []
}
```

Domain password available

```
{
  "ValidPropertiesMask": 1,
  "AutheticationMethods": [
    "cf189af5-01c5-469d-a859-a8f2f41ed153"
  ]
}
```

`authenticateByWindowsToken`

Authentication of the user using a Windows session token. The token is passed into the request via Windows authentication and Kerberos tickets.

```
[POST] /api/v5/logon/authenticateByWindowsToken
```

Request object

```
{
  "ApplicationId": "string",
  "BusinessApplication": "string"
}
```

in the request object:

- `ApplicationId` — required parameter. String identifier of the Axidian Access module.
- `BusinessApplication` — optional parameter. Name of the Axidian Access business application.

Response object

```
{
  "Token": "string",
  "LogonResult": {
    "Result": "string"
  },
  "UserId": "string"
}
```


Example of a successful response object


```
{
  "ValidPropertiesMask": 5,
  "Token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJleHAiOj.......",
  "LogonResult": null,
  "UserId": "UserId_10efa04f-7ba9-47d8-89db-56e166f1679f"
}
```

Policy

`get`

Returns information and settings of policies by identifier.

```
POST /api/v5/policy/get
```

Request object

```
{
  "Id": "00000000-0000-0000-0000-000000000000",
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `Id` — required parameter. Policy identifier.
- `AccessToken` — required parameter. Token of the administrator on whose behalf the API request is performed.
- `ApplicationId` — required parameter. String identifier of the Axidian Access module.

Response object

```
{
  "Id": "00000000-0000-0000-0000-000000000000",
  "Name": "string",
  "Description": "string",
  "Settings": \[
    {
      "Id": "00000000-0000-0000-0000-000000000000",
      "SettingType": 0,
      "Value": {},
      "ParentId": "00000000-0000-0000-0000-000000000000"
    }
  \],
  "CreatedById": "string",
  "CreatedTime": "2022-10-26T07:23:08.768Z",
  "LastEditedById": "string",
  "LastEditedTime": "2022-10-26T07:23:08.768Z"
}
```

in the response object:

- `SettingType`:
  - 0 — policy scope. The internal GUID of the scope objects is returned.
  - 1 — policy applications.
  - 2 — policy priority.

TemplateSession

`createTemplate`

Creates an authentication template by the identifier of an open session.

```
[POST] /api/v5/templateSession/createTemplate
```

Request object

```
{
 "SessionId": "00000000-0000-0000-0000-000000000000"
}
```

in the request object:

- `SessionId` — required parameter. The GUID of the open session obtained from the `openVerifySession` method is specified.

Response object

```
"00000000-0000-0000-0000-000000000000"
```

`openEnrollSession`

```
[POST] /api/v5/templateSession/openEnrollSession
```

`openVerifySession`

Opens a session for authentication of the user specified in `UserSearchParams` in the application `ApplicationId`, using the authentication method `ModeId`.

```
[POST] /api/v5/templateSession/openVerifySession
```

Request object

```
{
  "UserSearchParams": {
    "Id": "string",
    "Email": "string",
    "Phone": "string",
    "NameFormat": 0,
    "Name": "string",
    "ApplicationId": "string"
  },
  "ModeId": "00000000-0000-0000-0000-000000000000",
  "ApplicationId": "string"
}
```

in the request object:

- `UserSearchParams` — array of parameters for searching for the user;
  - `Id` — internal identifier of the user in Axidian Access;
  - `Email` — email of the user from the Active Directory catalog;
  - `Phone` — phone number of the user from the Active Directory catalog;
  - `NameFormat` — numeric value of the name format that will be specified in `Name`. The following formats are supported:
    - 0 — Undefined;
    - 1 — CanonicalName;
    - 2 — PrincipalName: *name@domain.name*;
    - 3 — SamCompatibleName: *domain\logon-name*;
    - 4 — DistinguishedName;
    - 5 — Sid;
- `ModeId` — required parameter. String identifier of the authentication method used.
- `ApplicationId` — required parameter. String identifier of the Axidian Access module.

Response object

```
GUID of the open session

"00000000-0000-0000-0000-000000000000"
```

`openIdentifySession`

```
[POST] /api/v5/templateSession/openIdentifySession
```

`prepareTemplateData`

Adds information about the user's template.

```
[POST] /api/v5/templateSession/prepareTemplateData
```

Request object

```
{
  "Data": {},
  "SessionId": "00000000-0000-0000-0000-000000000000"
}
```

in the request object:

- `Data` — the user's authentication data (a one-time code or password or other data depending on the authentication method used);
- `SessionId` — identifier of the open session obtained in `openVerifySession`.

Response object

```
{
  "EnoughData": true,
  "BinaryData": "string",
  "StringData": "string"
}
```

in the response object:

- `EnoughData` — indication that there is enough data from the client to build the template.

Example of a response object

```
{
  "ValidPropertiesMask": 1,
  "EnoughData": true,
  "BinaryData": null,
  "StringData": null
}
```

User

`searchUserId`

Returns `UserId` by the input parameters.

```
POST /api/v5/user/searchUserId
```

Request object

```
{
  "Id": "string",
  "Email": "string",
  "Phone": "string",
  "NameFormat": 0,
  "Name": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `NameFormat`:
  - 0 — Undefined,
  - 1 — CanonicalName,
  - 2 — PrincipalName,
  - 3 — SamCompatibleName,
  - 4 — DistinguishedName,
  - 5 — Sid.

Response object

```
"xxxx\_0000000-0000-0000-0000-000000000000"
```

UserCatalog

`getObjects`

Returns the data of an Axidian Access user catalog object by internal identifier.

```
POST /api/v5/userCatalog/getObjects
```

Request object

```
{
  "Ids": [
    "string"
  ],
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `Ids` — array of internal identifiers.
- `AccessToken` — required parameter. Token of the administrator on whose behalf the API request is performed.
- `ApplicationId` — required parameter. String identifier of the Axidian Access module.

Response object

```
{
  "Id": "string",
  "RawObjectId": "string",
  "Name": "string",
  "CanonicalName": "string",
  "PrincipalName": "string",
  "SamCompatibleName": "string",
  "DistinguishedName": "string",
  "Sid": "string",
  "IsGroup": true,
  "IsContainer": true,
  "IsRemoved": true,
  "GroupsIds": [
    "string"
  ],
  "ContainerId": "string"
}
```

`searchUsers`

Searches for a user by the input parameters.

```
POST /api/v5/userCatalog/searchUsers
```

Request object

```
{
  "Operation": 0,
  "Filters": [
    {
      "AttributeName": "string",
      "Value": {},
      "Negation": true
    }
  ],
  "Limit": 0,
  "LoadParentObjects": true,
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `Operation`:
  - 0 — OR;
  - 1 — AND;

- `Filters` - array of filters for the search:
  - `AttributeName` — name of the Active Directory attribute to search by;
  - `Value` — value of the Active Directory attribute to search for;
  - `Negation` — negation of the specified filter. Values *true\false*.

Response object


```
[
  {
    "FirstName": "string",
    "MiddleName": "string",
    "LastName": "string",
    "Email": "string",
    "Phone": "string",
    "IsDisabled": true,
    "IsLocked": true,
    "Id": "string",
    "RawObjectId": "string",
    "Name": "string",
    "CanonicalName": "string",
    "PrincipalName": "string",
    "SamCompatibleName": "string",
    "DistinguishedName": "string",
    "Sid": "string",
    "IsGroup": true,
    "IsContainer": true,
    "IsRemoved": true,
    "GroupsIds": [
      "string"
    ],
    "ContainerId": "string"
  }
]
```


`searchGroups`

Searches for a group by the input parameters.

```
POST /api/v5/userCatalog/searchGroups
```

Request object

```
{
  "Operation": 0,
  "Filters": [
    {
      "AttributeName": "string",
      "Value": {},
      "Negation": true
    }
  ],
  "Limit": 0,
  "LoadParentObjects": true,
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `Operation`:
  - 0 — OR;
  - 1 — AND;
- `Filters` — array of filters for the search:
  - `AttributeName` — name of the Active Directory attribute to search by;
  - `Value` — value of the Active Directory attribute to search for;
  - `Negation` — negation of the specified filter. Values *true\false*.

Response object

```
[
  {
    "FirstName": "string",
    "MiddleName": "string",
    "LastName": "string",
    "Email": "string",
    "Phone": "string",
    "IsDisabled": true,
    "IsLocked": true,
    "Id": "string",
    "RawObjectId": "string",
    "Name": "string",
    "CanonicalName": "string",
    "PrincipalName": "string",
    "SamCompatibleName": "string",
    "DistinguishedName": "string",
    "Sid": "string",
    "IsGroup": true,
    "IsContainer": true,
    "IsRemoved": true,
    "GroupsIds": [
      "string"
    ],
    "ContainerId": "string"
  }
]
```


`searchContainers`

Searches for an organizational unit by the input parameters.

```
POST /api/v5/userCatalog/searchContainers
```

Request object

```
{
  "Operation": 0,
  "Filters": [
    {
      "AttributeName": "string",
      "Value": {},
      "Negation": true
    }
  ],
  "Limit": 0,
  "LoadParentObjects": true,
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

in the request object:

- `Operation`:
  - 0 — OR;
  - 1 — AND;
- `Filters` — array of filters for the search:
  - `AttributeName` — name of the Active Directory attribute to search by;
  - `Value` — value of the Active Directory attribute to search for;
  - `Negation` — negation of the specified filter. Values *true\false*.

Response object

```
[
  {
    "FirstName": "string",
    "MiddleName": "string",
    "LastName": "string",
    "Email": "string",
    "Phone": "string",
    "IsDisabled": true,
    "IsLocked": true,
    "Id": "string",
    "RawObjectId": "string",
    "Name": "string",
    "CanonicalName": "string",
    "PrincipalName": "string",
    "SamCompatibleName": "string",
    "DistinguishedName": "string",
    "Sid": "string",
    "IsGroup": true,
    "IsContainer": true,
    "IsRemoved": true,
    "GroupsIds": [
      "string"
    ],
    "ContainerId": "string"
  }
]
```

UserProfile

`findLogonInfoByUser`

Returns information about the user's authentications.

```
POST /api/v5/userProfile/findLogonInfoByUser
```

Request object

```
{
  "UserId": "string",
  "AccessToken": "string",
  "ApplicationId": "string"
}
```

Response object

```
{
  "UserId": "xxxx\_000000-0000-0000-0000-00000000000",
  "ApplicationId": "Native Enroller",
  "LogonDate": "2022-05-04T10:54:05.821+03:00",
  "ModeId": "000000-0000-0000-0000-00000000000"
}
```

in the response object:

- `UserId` — internal identifier of the user;
- `ApplicationId` — identifier of the application that was logged into;
- `LogonDate` — date and time of login;
- `ModeId` — identifier of the login method.

Use cases

Create reports

Create reports

Collect statistics on user enrolled authenticators

Collect statistics on user enrolled authenticators

Retrieve the list of users with licenses

Retrieve the list of users with licenses

Two-factor authentication in the API

Setting up two-factor authentication via the API

Remove users from a policy

Remove users from a policy

Retrieve the list of users from a policy

Retrieve the list of users from a policy

Create reports

> **INFORMATION**
> You can download the script for creating reports from this link.

Prerequisites

- The Active Directory module for PowerShell installed on Windows Server.
- A working Axidian Access server.
- The user on whose behalf the script will be run must have the minimal global *Inspector* rights in the Axidian Access system.

How the script works

The script generates reports according to prepared scenarios. The Axidian Access Core API and Axidian Access Log Server are used to create the reports.

The script supports configuring filtering for generating a report:

- selection from a user catalog container,
- selection by a specific user,
- selection by date.

Filters

The *Container* field

The field is intended for selecting the organizational units in the domain in which the search for users will be performed. The value *Entire Catalog* means that the root catalog with the user catalog used by the Axidian Access server is selected.

The search for organizational units from the user catalog is performed using the `searchContainers` method in the corresponding function in the file *Base/AMAPI/UserCatalog.ps1*.

> **NOTE**
> The export of containers is performed from the user catalog. The value in the field is not editable.
>
> The name of the root user catalogs specified in the Core Server configuration file is not displayed.

To select the required organizational unit from the catalog, do the following:

1. Click the **Select** button.
2. In the drop-down list of the **Containers** window, select a container and click **Ok**.

The *Groups* field

The field is intended for searching for groups in the domain or in the organizational unit that was selected in the *Container* field. Default value *Not set*.

The search for the user is performed using the `searchGroups` method in the corresponding function in the file *Base/AMAPI/UserCatalog.ps1*.

The preparation of the request using the specified filters is performed in the `searchGropsUseFilters` function in the file *Scenarios/Reports/SearchUseFilters.ps1*.

The Active Directory `Name` parameter is taken into account in the search.

To search for a group, select a container if necessary and enter the group name. If no data is specified, then all groups from the user catalog will be found.

The *User or initiator* field

The field is intended for searching for a user in the domain; in the organizational unit that was selected in the *Container* field; the group from the *Groups* field. Default value *Not set*.

The search for the user is performed using the `searchUsers` method in the corresponding function in the file *Base/AMAPI/UserCatalog.ps1*.

The preparation of the request using the specified filters is performed in the `searchUsersUseFilters` function in the file *Scenarios/Reports/SearchUseFilters.ps1*.

The following parameters are taken into account in the search:

- `ContainerId` — identifier of the container in which the search for the user will be performed. The parameter is used if an organizational unit was selected in the *Container* field.
- `FirstName`, `MiddleName`, `LastName`, `PrincipalName`, `Name` — the corresponding values of the user's attributes from Active Directory.

> ⓘ **NOTE**
> All parameters are used with the logical operation *OR*.

To search for users, you can click the **Select** button and select all the users found.

To search for a user, you can specify part of the name; at the end of the specified part, add the * character.

Configuration

For the script to work, you need to configure the *Config.ps1* file, which is located in the root of the folder with the script.

> ⚠ **IMPORTANT!**
> This section describes the general configuration of the script. Individual reports may require additional configuration.

In the file:

- `$domainName` — the domain name is specified. Required parameter.
- `$eventsCount` — limit on the export of events. Optional parameter. By default, the number of events received is not limited, the parameter is commented out.
- `$serverUrl` — URL of the Axidian Access server. Required parameter.
- `$logServerUrl` — URL of the Axidian Access Log Server. Required parameter.
- `$logsOnFile` — enabling\disabling logging to a file. Required parameter. Enabled by default.
- `$logsOnConsole` — enabling\disabling logging to the console. Required parameter. Disabled by default.
- `$apiVersion` — version of the API used in the script. Required parameter.

Creating a report with the number of authentications by the specified providers

In this report, a table is generated with the number of a user's authentications by an array of authenticators in a specific Axidian application, specified in the *Config.ps1* file.

The creation of the report is performed in the `createLogonsByProviderReport` function in the file *Create-reports/Scenarios/Reports/CreateReports.ps1*.

Configuration

> ⚠ **IMPORTANT!**
> This section describes the settings for a specific report. Before starting work, you need to perform the basic configuration described in the Configuration section.

To configure, open the *Config.ps1* file and do the following:

1. In the `AuthForReport` variable, specify the identifier of the Axidian provider and the name for the report:
   - The values must be specified in the format *Axidian provider GUID=Name for the report*.
   - The GUID values of the providers can be found in the Authentication method identifiers section.
   - The name for the report can be arbitrary. This name will be used when creating the table.

   Example of configuration with domain password and fingerprint

   ```
   $AuthForReport          = @{
       'CF189AF5-01C5-469D-A859-A8F2F41ED153' = 'Password'
       'A0EF00AD-1EEB-4D48-8BCF-06E19CD5585F' = 'Fingerprint'
   }
   ```

2. In the `ApplicationId` variable, the name of the Axidian integration module is specified. The list of modules can be found in the Application identifiers section.

   Example of configuration for the Windows Logon application

   ```
   $ApplicationId="Windows Logon"
   ```

3. You can specify the format in which the report will be exported. CSV and HTML formats are supported. You can export the report in two formats simultaneously. If both variables are undefined, this will cause an export error.

   To export the report to CSV, set the `CsvReport` variable to *true*.

   ```
   $CsvReport = 'true'
   ```

   To export the report to HTML, set the `HtmlReport` variable to *true*.

   ```
   $HtmlReport = 'true'
   ```

4. In the `pathReportLogonsByProvider` variable, the path to the folder where the report will be saved is specified. By default, the root of the folder with the script is used.

   ```
   $pathReportLogonsByProvider="$ScriptDirectory\Reports\LogonsByProvider"
   ```

Example of execution

1. To start working with the script, open the PowerShell snap-in, go to the folder with the script and run the *Start.ps1* file via the `.\Start.ps1` command.
2. If necessary, configure the script filters and click the **Report of number of logins by providers** button.
3. After the script finishes, an HTML-format table will be generated at the path specified during configuration.

Collect statistics on user enrolled authenticators

> **INFORMATION**
> You can download the script for collecting statistics from this link.

Prerequisites

- The Active Directory module for PowerShell installed on Windows Server.
- A working Axidian Access server.
- The user on whose behalf the script will be run must have the minimal global *Inspector* rights in the Axidian Access system.

How the script works

The operation of the script consists of several stages:

1. The script iterates over the policy identifiers from the array in the configuration file *Config.ps1*.
2. Using the `/api/v5/policy/get` method, information about the policy is requested.
3. From the scope property, information about the configured objects is requested using the `/api/v5/userCatalog/getObjects` method.
4. From the obtained objects, users are requested:
   - Using the PowerShell cmdlet `Get-ADGroup` in the function `getUsersCN` of the file *Base/AdditionalFunctions/GetUsers.ps1* for an Active Directory group.
   - Using the PowerShell cmdlet `Get-ADUser` in the function `getUsersOU` of the file *Base/AdditionalFunctions/GetUsers.ps1* for an Active Directory organizational unit.
5. For the obtained users, a report is compiled on the state of the authenticators specified in the `AuthForReport` variable of the file *Config.ps1*.
   - Using the `/api/v5/authenticator/findByUserIds` method, information with all the internal GUIDs of the user's authenticators is requested.
   - Using the `/api/v5/authenticator/getUserAuth` method, information for a specific GUID is requested.
6. Using the `addDataToCSV` function of the file *Base/AdditionalFunctions/CsvReport.ps1* a table in CSV format is assembled.

Configuration

For the script to work, you need to configure the file *Config.ps1*, which is located in the root of the folder with the script.

In the file:

- `$serverUrl` — URL of the Axidian Access server. Required parameter.
- `$logsOnFile` — enabling\disabling logging to a file. Required parameter. Enabled by default.
- `$logsOnConsole` — enabling\disabling logging to the console. Required parameter. Disabled by default.
- `$apiVersion` — version of the API used in the script. Required parameter. Default value: *v5*.
- `$policyIds` — policy identifiers, from whose scope the users are requested.
- `$AuthForReport` — this variable specifies the GUIDs of the Axidian authenticators and their name for the report. The name can be arbitrary. The values are specified in the following format: `'GUID' = "Name"`.

  Example of configuration for Axidian Key and Passcode

  ```
  $AuthForReport      = @{
      'F696F05D-5466-42b4-BF52-21BEE1CB9529' = 'Passcode'
      'DEEF0CB8-AD2F-4B89-964A-B6C7ECA80C68' = 'Push notification in Axidian Key'
  }
  ```

- `$csvPathToReport` — path to the CSV report file. By default the file is created in the script folder at the path *Reports\Statistics*.

Example of execution

1. To run the script, execute the file *main.ps1*.
2. During the execution of the script, a CSV-format file will be created at the path from the `$csvPathToReport` variable.

> **NOTE**
> Users are added to the report sequentially.

Retrieve the list of users with licenses

> **INFORMATION**
>
> You can download the script for retrieving the list of users from this link.

Prerequisites

- The Active Directory module for PowerShell installed on Windows Server.
- A working Axidian Access server.
- The user on whose behalf the script will be run must have the minimal global *Inspector* rights in the Axidian Access system.

How the script works

The script generates a CSV-format table with the following columns:

- <UserName>,
- <true>,
- <false>,

where *true**false* are the columns with the corresponding license name.

The array of licenses whose presence is checked for the user is the list of all registered licenses.

The operation of the script consists of several stages:

1. An array of licenses loaded into the system is generated using the `/license/getAllLicenses` method.
2. Using the `/api/v5/policy/get` method, information about the policy is requested.
3. From the scope property, information about the configured objects is requested using the `/api/v5/userCatalog/getObjects` method.
4. From the obtained objects, users are requested:
    - Using the PowerShell cmdlet `Get-ADGroup` in the function `getUsersCN` of the file *Base/AdditionalFunctions/GetUsers.ps1* for an Active Directory group.
    - Using the PowerShell cmdlet `Get-ADUser` in the function `getUsersOU` of the file *Base/AdditionalFunctions/GetUsers.ps1* for an Active Directory organizational unit.

5. Using the `license/getCatalogObjectLicenses` method, the presence of the user's licenses is requested and compared with the overall array of licenses.
6. Using the `addDataToCSV` function of the file *Base/AdditionalFunctions/CsvReport.ps1* a table in CSV format is assembled.

Configuration

For the script to work, you need to configure the *Config.ps1* file, which is located in the root of the folder with the script.

In the file:

- `$serverUrl` — URL of the Axidian Access server. Required parameter.
- `$logsOnFile` — enabling\disabling logging to a file. Required parameter. Enabled by default.
- `$logsOnConsole` — enabling\disabling logging to the console. Required parameter. Disabled by default.
- `$apiVersion` — version of the API used in the script. Required parameter. Default value: *v5*.
- `$policyIds` — policy identifiers, from whose scope the users are requested.
- `$csvPathToReport` — path to the CSV report file. By default the file is created in the script folder at the path *Reports\Statistics*.

Example of execution

1. To run the script, execute the file *main.ps1*.
2. During the execution of the script, a CSV-format file will be created at the path from the `$csvPathToReport` variable.

Two-factor authentication in the API

> **INFORMATION**
>
> You can download the script for performing two-factor authentication via the API from this link.

How the script works

The script performs authentication via the standard Core Server API. Providers with a one-time password, Passcode, and Windows Password are supported as authentication providers.

To perform authentication via the API, do the following:

1. Call the `/api/v5/templateSession/openVerifySession` method. On successful execution of the method, a session identifier is returned — the `$SessionId` variable in the script.

   > **NOTE**
   >
   > As an example, the script uses the *PrincipalName* name format.

2. The following steps differ depending on the authentication provider used:

If the one-time code is known

If the one-time code is known and does not need to be sent, then authentication is performed using the following algorithm (the `SimpleLogon` function in the script):

1. Input of the one-time code is requested.
2. Preparation of the authentication template — the `/api/v5/templateSession/prepareTemplateData` method is called.

The one-time code is passed as `Data` as a string (line #12 in the script). If the preparation was successful, then a template is created and the `/api/v5/templateSession/createTemplate` method is called.

The final step is authentication of the user using the `/api/v5/logon/authenticate` method.

On successful authentication, a token is returned in JSON format.

If the one-time code is not known

If the one-time code is not known, then you first need to initiate its sending using the `LogonByOtp` function in the script.

1. Call the `/api/v5/templateSession/prepareTemplateData` method. Initially, the value *Null* is passed as `Data`.

   The first call is necessary so that the server sends the one-time code, for example sending an SMS. Also, on the initial request the value of the `EnoughData` parameter will contain *false*.

2. Input of the one-time code is requested.

3. Preparation of the authentication template — the `/api/v5/templateSession/prepareTemplateData` method is called.

The one-time code is passed as `Data` as a string (line #12 in the script). The request will return an object where the value of the `EnoughData` parameter will contain *true*. If the preparation was successful, then a template is created and the `/api/v5/templateSession/createTemplate` method is called.

The final step is authentication of the user using the `/api/v5/logon/authenticate` method. On successful authentication, a token is returned in JSON format.

Prerequisites

- a working Axidian Access server,
- the presence of an *Authentication API* license.

Configuration

For the script to work, you need to configure the *Config.ps1 file*, which is located in the root of the folder with the script.

`$serverUrl` — URL of the Axidian Access server. Required parameter. `$logsOnFile` — enabling\disabling logging to a file. Required parameter. Enabled by default. `$logsOnConsole` — enabling\disabling logging to the console. Required parameter. Disabled by default. `$apiVersion` — version of the API used in the script. Required parameter. Default value — *v5*.

In the main file *main.ps1* do the following:

1. In the `$ProviderGuid` parameter, specify the GUID of the authentication provider that will be used in the script.

   As an example, the script uses the identifier of the Axidian Access SMS OTP Provider — {EBB6F3FA-A400-45F4-853A-D517D89AC2A3}.

2. In the `$UserName` parameter, specify the user name in UPN format.

Remove users from a policy

> **INFORMATION**
>
> You can download the script for collecting statistics from this link.

Prerequisites

- A working Axidian Access server.
- The presence of an Authentication API license.

How the script works

The script removes a user from the specified policy using the standard Axidian Access Core API.

Configuration

Parameter	What it is responsible for	Requirement	Default settings
$serverUrl	Contains the URL of the Axidian Access server	Required parameter	None
$logsOnFile	Enabling\disabling logging to a file	Required parameter	Enabled by default
$logsOnConsole	Enabling\disabling logging to the console	Required parameter	Disabled by default
$apiVersion	Version of the API used in the script	Required parameter	Default value — v6

Example of execution

1. Run the *main.ps1* script.

2. Enter the user name in *UPN* format. A list of the numbers of all policies that the user belongs to will be generated.
3. Enter the numbers of the policies from which you want to remove the user, separated by commas (or specify only one policy).

Example

```
Enter UPN: Admin-Axidian@axidian.local
User is in scope of the following policies:
0. Radius
1. Main
2. WL + ESSO
Choose policies to remove user from (e.g. 0,1,2): 0,2
User successfully removed from Radius
User successfully removed from WL + ESSO
```

Retrieve the list of users from a policy

> **INFORMATION**
>
> You can download the script for retrieving the list of users from an Axidian policy from this link.

Prerequisites

- The Active Directory module for PowerShell installed on Windows Server.
- A working Axidian Access server.
- The user on whose behalf the script will be run must have the minimal global *Inspector* rights.

How the script works

The script exports the list of users from the specified policy in CSV format. The user attributes *UserPrincipalName* and *displayName* are exported.

1. The script iterates over the policy identifiers from the array in the configuration file *Config.ps1*.
2. Using the `/api/v5/policy/get` method, information about the policy is requested.
3. From the scope property, information about the configured objects is requested using the `/api/v5/userCatalog/getObjects` method.
4. From the obtained objects, users are requested:
   - Using the `Get-ADGroup` cmdlet in the `getUsersCN` function of the file *Base/AdditionalFunctions/GetUsers.ps1* for an Active Directory group.
   - Using the `Get-ADUser` cmdlet in the `getUsersOU` function of the file *Base/AdditionalFunctions/GetUsers.ps1* for an Active Directory organizational unit.
   - Using the `Get-ADUser` cmdlet in the `getADUser` function of the file *Base/AdditionalFunctions/GetUsers.ps1* for an Active Directory user.
5. Using the `addDataForUserInPolicyReportCsv` function of the file *Scenarios/Reports/CsvReport.ps1*, a table in CSV format is assembled.

Configuration

1. In the file *Config.ps1*, in the `$serverUrl` variable, specify the address of the Axidian Access server in the format `https://axidian.access.local/`.

2. In the `$logServerUrl` variable, specify the address of the Log Server in the format `https://axidian.access.local/`.
3. In the `$policyIds` variable, specify the identifier of the target policy in the format *@('2fc7eeec-5adf-4a50-bad3-ba04585c76cd')*. You can specify several separated by a comma *@('2fc7eeec-5adf-4a50-bad3-ba04585c76cd','7436daa7-abf7-437b-991f-0e89cc9f1b9b')*.

> **TIP**
> You can find the policy identifier in the Management Console. Open the desired policy; the identifier is contained in the browser address bar in the `policyId` variable.

The file with the exported users is saved in the directory *$ScriptDirectory\Reports\UserInPolicyReport*. If you want to save the file in another location, specify the path in the `$pathReportUserInPolicyReport` variable in the file *Config.ps1*.

To run the script, execute the file *Start.ps1* using PowerShell.

Troubleshooting

Server component logs

Logs of Core Server, Log Server, Key Server, Management Console, User Console, and IDP

GetLog

Logs of IIS Extension, Radius Extension, RDP Windows Logon, Windows Logon, and ESSO

Technical support

How to contact Axidian technical support

Server component logs

Log levels

Depending on how detailed the information about the component operation must be, you can set different log levels. They define how important and detailed the information written to the log files is. This allows you to filter and analyze the logs more efficiently.

We recommend that you use the *Trace* log level as the most informative one.

Log levels

Trace	The most detailed level. Logs with the Trace level contain all the information about the component operation processes, including the details of API method calls.
Debug	At this log level, the records contain the details of the component operation, significant variables, and other data that may be useful for detecting and fixing errors.
Info	At this log level, informational messages notifying about the normal functioning of the component are written. They may include such events as starting or completing processes, editing a user profile, and others.
Warn	This log level is used to write warnings and notifications about potential errors and abnormal situations. The events are not critical but require attention. The component can continue working.
Error	The log level used to write errors that led to incorrect component operation or serious problems. Logs with the Error level indicate problems that require intervention and correction.
Fatal	The least detailed log level. If this level is set, only the most critical errors and problems that lead to the immediate termination of the component or other serious consequences are written. Logs with the Fatal level usually indicate serious failures that require immediate intervention and correction.

Collect Core Server logs

Enable logging

1. Use an administrator account to open the `C:\inetpub\wwwroot\am\core\Config\nlog.config` file.
2. In the `logger` tags, set the `minlevel` parameter to `Trace` and the `enabled` parameter to `true`.

Example

```
<rules>
  <logger name="*" writeTo="f1" minlevel="Trace" enabled="true" >
    <filters>
      <when condition="equals('${logger}','metricTime')" action="Ignore" />
      <when condition="equals('${logger}','metricInfo')" action="Ignore" />
    </filters>
  </logger>
  <logger name="metricTime" writeTo="metricTimeTarget" minlevel="Trace" enabled="true" />
  <logger name="metricInfo" writeTo="metricInfoTarget" minlevel="Trace" enabled="true" />
</rules>
```

3. Save the file and restart the IIS server.

Collect logs

1. Delete the existing Core Server logs in the `C:\inetpub\wwwroot\am\core\Logs` folder.
2. Reproduce the problem.
3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect Management Console logs

Enable logging

1. Use an administrator account to open the `C:\inetpub\wwwroot\am\mc\Config\nlog.config` file.
2. In the `logger` tag, set the `minlevel` parameter to `Trace` and the `enabled` parameter to `true`.

Example

```
<rules>
    <logger name="*" writeTo="f1" minlevel="Trace" enabled="true" />
</rules>
```

3. Save the file and restart the IIS server.

Collect logs

1. Delete the existing Management Console logs in the `C:\inetpub\wwwroot\am\mc\Logs` folder.
2. Reproduce the problem.
3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect Log Server logs

Enable logging

1. Use an administrator account to open the `C:\inetpub\wwwroot\ls\nlog.config` file.
2. In the `logger` tag, set the `minlevel` parameter to `Trace`. Create the `enabled` parameter and set it to `true`.

   Example

   ```
   <rules>
       <logger name="*" minlevel="Trace" writeTo="file" enabled="true" />
   </rules>
   ```

3. Save the file and restart the IIS server.

Collect logs

1. Delete the existing Log Server logs in the `C:\inetpub\wwwroot\ls\Logs` folder.
2. Reproduce the problem.
3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect Key Server logs

Enable logging

1. Use an administrator account to open the `C:\inetpub\wwwroot\axidiankey\Web.Config` file.
2. In the `logger` tag, set the `minlevel` parameter to `Trace` and the `enabled` parameter to `true`.

Example

```
<rules>
   <logger name="*" minlevel="Trace" writeTo="fileLog" enabled="true" />
</rules>
```

3. Save the file and restart the IIS server.

Collect logs

1. Delete the existing Axidian Key server logs in the `C:\inetpub\wwwroot\axidiankey\logs` folder.
2. Reproduce the problem.
3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect User Console logs

Enable logging

1. Use an administrator account to open the `C:\inetpub\wwwroot\am\uc\Config\nlog.config` file.
2. In the `logger` tag, set the `minlevel` parameter to `Trace` and the `enabled` parameter to `true`.

Example

```
<rules>
   <logger name="*" writeTo="f1" minlevel="Trace" enabled="true" />
</rules>
```

3. Save the file and restart the IIS server.

Collect logs

1. Delete the existing User Console logs in the `C:\inetpub\wwwroot\am\uc\logs` folder.
2. Reproduce the problem.
3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect Identity Provider logs

Enable logging

1. Use an administrator account to open the following file:
   - for Axidian Access 8.1.5 and higher — `C:\inetpub\wwwroot\am\idp\nlog.config`
   - for Axidian Access 8.1.4 and lower — `C:\inetpub\wwwroot\am\idp\Config\nlog.config`
2. In the `logger` tag, set the `minlevel` parameter to `Trace` and the `enabled` parameter to `true`.

   Example

   ```
   <rules>
      <logger name="*" writeTo="f1" minlevel="Trace" enabled="true" />
   </rules>
   ```

3. Save the file and restart the IIS server.

Collect logs

1. Delete the existing Identity Provider logs in the `C:\inetpub\wwwroot\am\idp\logs` folder.
2. Reproduce the problem.
3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect Mobile Device Provisioning logs

Enable logging

1. Open the `C:\inetpub\wwwroot\am\mdp\nlog.config` configuration file.
2. In the `rules` section, set the `minlevel` parameter to `Trace` and the `enabled` parameter to `true`.

   Example

   ```
   <rules>
      <logger name="*" writeTo="f1" minlevel="Trace" enabled="true" />
   </rules>
   ```

3. Save the changes.

Collect logs

1. Delete the existing Mobile Device Provisioning logs in the `C:\inetpub\wwwroot\am\mdp\logs` folder.
2. Reproduce the problem.
3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect cache server logs

> **INFORMATION**
>
> The cache server logs are collected on the client machines with the Windows Logon or ESSO Agent components installed.

Enable logging

1. Use an administrator account to open the `C:\Program Files (x86)\Common Files\Axidian-Id\AM\CacheServer\NLog.config` file for 64-bit machines or `C:\Program Files\Common Files\Axidian-Id\AM\CacheServer\NLog.config` for 32-bit machines.
2. In the `logger` tags, set the `enabled` parameter to `true`.

   Example

   ```
   <rules>
        <logger name="*" minlevel="Trace" writeTo="f1" encoding="Unicode" enabled="true" >
         <filters>
            <when condition="equals('${logger}','metricTime')" action="Ignore" />
            <when condition="equals('${logger}','metricInfo')" action="Ignore" />
         </filters>
      </logger>
     <logger name="*" minlevel="Trace" writeTo="console" encoding="Unicode" enabled="true" />
     <logger name="metricTime" writeTo="metricTimeTarget" minlevel="Trace" enabled="true" />
     <logger name="metricInfo" writeTo="metricInfoTarget" minlevel="Trace" enabled="true" />
   </rules>
   ```

3. Save the changes in the file.

Collect logs

1. Delete the existing cache server logs in the `C:\Windows\System32\LogFiles\Axidian-ID Cache` folder.
2. Reproduce the problem.

3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect configuration wizard logs

Enable logging

1. Use an administrator account to open the `C:\Program Files\Axidian\Wizard\NLog.config` file.
2. In the `logger` tag, set the `minlevel` parameter to `Trace` and the `enabled` parameter to `true`.

Example

```
<rules>
   <logger name="*" writeTo="file" minlevel="Trace" enabled="true" />
</rules>
```

3. Save the file and restart the configuration wizard.

Collect logs

1. Delete the existing configuration wizard logs in the `C:\Program Files\Axidian\Wizard\Logs` folder.
2. Reproduce the problem.
3. Put together an archive with the logs and send it to Axidian Technical Support along with a detailed description of the user activity and the exact time when the problem was reproduced.

Collect Telegram Service logs

Enable logging

1. Use an administrator account to open the `C:\Program Files (x86)\Axidian-Id\Axidian-Id Telegram Service\Nlog.config` file.
2. In the `logger` tag, set the `minlevel` parameter to `Trace` and the `enabled` parameter to `true`.

Example

```
<rules>
   <logger name="*" writeTo="f1" minlevel="Trace" encoding="Unicode" enabled="true">
   </logger>
</rules>
```

3. Save the changes in the file and restart the Telegram Service.

Collect logs

1. Delete the existing Telegram Service logs in the `C:\Program Files (x86)\Axidian-Id\Axidian-Id Telegram Service\Logs` folder.
2. Reproduce the problem.

GetLog

The Axidian Access GetLog application is designed to collect software logs of the following components locally and remotely:

- IIS Extension
- Radius Extension
- RDP Windows Logon
- Windows Logon
- Enterprise Single Sign-On

Connect to a computer

To connect to the computer whose logs you need to get:

1. Run the GetLog utility as a local administrator. The executable file is located at `Axidian Access <version number>\logging\AxidianAccess.GetLog.exe`.

2. In the **Computer** field, enter the name or IP address of the remote computer. To connect to a local computer, enter *localhost* or *127.0.0.1*.

3. Click **Connect**. After the connection is established, the following options become available:

    - The **Enable Log/Disable Log** button for enabling and disabling logs
    - The **Get Log** button for getting logs
    - The **Advanced Settings** button for switching to the advanced settings
    - The **Disconnect** button

> **TIP**
> To connect to a remote computer running Windows 7 or higher, make sure that the **Windows Management Instrumentation (WMI)** service is running and is not disabled on the remote computer.

Enable/disable and collect logs

Using the utility

To enable or disable logging and get the logs:

1. Connect to the computer.
2. To enable logging, click **Enable Log**.
3. If logging is required to reveal the root cause of a problem, perform the actions required to reproduce the problem.
4. To disable logging, click **Disable Log**.
5. To get the logs, click **Get Log...**.
6. Specify the folder to save the ZIP archive to and click **Save**.
7. To disconnect from the computer, click **Disconnect** and close the application.

Using the registry

In some cases, you may need to enable or disable logging manually. To do this:

1. Open Registry Editor.
2. Go to the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\Logging` directory.
3. Set the value of the `Enabled` parameter:
   - 0 to disable logging
   - 1 to enable logging

> **NOTE**
>
> By default, the logs are written to the `\WINDOWS\System32\LogFiles\Axidian-Id` directory. To access the logs of a remote computer, the `ADMIN$\System32\LogFiles\Axidian-Id` network directory is used by default. You can specify the directory for writing logs in Advanced Settings > Use alternative location.
>
> If the error *The system cannot find the file specified* appears when you try to save the logs, make sure that the `C:\Windows\System32\LogFiles\Axidian-ID` folder exists. Create the folder manually if it is missing.

Advanced settings

Settings in the utility

To switch to the advanced settings, click **Advanced Settings** in the main window of Axidian Access GetLog. The following settings are available in the **Advanced Settings** window:

- **Max. log size (bytes)** — the maximum size of all files in the directory, in bytes. The default value is 1 GB. When the specified value is reached, all files in the directory whose modification date is older than the value of the *Max. log file age* field are deleted.
- **Max. log file age (s)** — the log file age in seconds. If the log size in the directory exceeds the *Max. log size* value, all files in the directory whose modification date is older than the value of this field are deleted.
- **Cleaner interval (s)** — the interval of the log directory size check, in seconds. The default value is 1 hour (3600 seconds).
- **Activity checking period (ms)** — the interval of the logging activity check, in milliseconds. Before starting to write logs, the Axidian Access component checks whether logging is enabled at the workstation. By default, the check interval is 1 minute (60,000 milliseconds).
- **Enable log cycling** — the cyclic log writing mode. If the option is enabled, the logs of each process are written according to the specified settings for the file number and size.
  - **Max. size of a log file (bytes)** — the maximum log size in bytes. The default value is 10 MB (1000000 bytes). When the specified size is reached, the file contents is overwritten with new data.
  - **Max. number of saved log files** — the maximum number of saved logs. The default value is 5, not counting the file currently being written. If the specified number of files is exceeded, the oldest one is deleted and writing continues to a newly created file.
- **Use alternative location** — an alternative directory for writing logs. If the option is disabled, the logs are written to the default directories:
  - `ADMIN$\System32\LogFiles\Axidian-Id` — the network path
  - `\WINDOWS\System32\LogFiles\Axidian-Id` — the local path

Settings in the registry

To switch to the advanced settings in the registry, open Registry Editor and go to the `HKEY_LOCAL_MACHINE\SOFTWARE\Axidian-ID\Logging` directory. The following parameters are available in the registry:

- `CycleLogsFileCountMax` — the maximum number of saved logs. The default value is 5, not counting the file currently being written. If the specified number of files is exceeded, the oldest one is deleted and writing continues to a newly created file.
- `CycleLogsFileSizeMaxBytes` — the maximum log size in bytes. The default value is 10 MB (1000000 bytes). When the specified size is reached, the file contents is overwritten with new data.
- `EnabledRecheckPeriodMs` — the interval of the logging activity check, in milliseconds. Before starting to write logs, the Axidian Access component checks whether logging is enabled at the workstation. By default, the check interval is 1 minute (60,000 milliseconds).
- `MaxLogFileAge` — the log file age in seconds. If the log size in the directory exceeds the `MaxLogSize` value, all files in the directory whose modification date is older than the value of this parameter are deleted.
- `MaxLogSize` — the maximum size of all files in the directory, in bytes. The default value is 1 GB. When the specified value is reached, all files in the directory whose modification date is older than the value of the `MaxLogFileAge` parameter are deleted.
- `RootLogPath` — an alternative directory for writing logs. If the parameter is disabled, the logs are written to the default directories:
  - `ADMIN$\System32\LogFiles\Axidian-Id` — the network path
  - `\WINDOWS\System32\LogFiles\Axidian-Id` — the local path

Technical support

If you have questions regarding the product documentation or you are experiencing an issue with the functionality of your Axidian Access solution, you can contact Axidian technical support.

Follow these steps to submit a support request.

1. Open Technical Support Portal.
2. Enter your email address and password and click **Login**.
3. Click **Submit a Ticket**.
4. Select department and click **Next**.
5. Fill in the required fields and click **Submit**.

What's new

This section contains an overview of updates and improvements in Axidian Access 8.2.

8.2.12

- Corrected the processing of the license expiration date when a user signs in to a licensed application.
- In NPS RADIUS Extension, optimized the processing of duplicate authentication requests at session termination.

8.2.9–8.2.11

Minor improvements and bug fixes.

8.2.8

- Improved the operation of Management Console 8.x on Axidian Access 9.x database.
- For Secured TOTP provider, added support for caching.
- Fixed a defect that prevented successful installation of Windows Logon without internet connection.
- Fixed an error during registration and password change for Active Directory in User Console.

8.2.7

- Updated distribution package contents (Telegram provider).
- Fixed errors in signatures of some MSI packages.

8.2.6

- Added the option to send OTP using a script through Messages Proxy web application.
- For Windows Password provider, added the functionality that allows configuring the method of credentials verification on domain controller.

- In Radius Extension, added the ability to write provider chain in one string – One-string authenticator.
- Added new authentication method for Radius – Passcode + SMS.
- In Windows Logon and RDP Windows Logon, added support for Secured TOTP.
- Added a setting that allows changing the deeplink for Axidian Key and Secured TOTP registration and Axidian Key application download.
- Changed the process of Axidian Key authenticator registration via link from email.
- Added the ability to copy a six-digit code to the clipboard from the Axidian Key mobile application.
- Modified the format of the license validity start and end time.
- Improved the MSI package installation process in an isolated environment.
- Expanded capabilities for application interaction via OpenID Connect protocol.

8.2.4

- Added a utility that can be used to generate keys for user token encryption during manual configuration of Core Server.
- Added the functionality that can be used to search for and delete objects from the policy scope.
- Added support for the Windows security policy *Interactive logon: Smart card removal behavior* in multi-factor authentication where smart cards are used as the second factor.
- Improved the operation of NPS Radius Extension with logging enabled.
- Fixed an error related to repeated authentications through NPS Radius Extension.

8.2.3

- Fixed the authentication error that occurred because user search in the catalog took a long time.
- Fixed several errors related to duplicate requests in NPS Radius Extension.
- Fixed the authentication error in NPS Radius Extension, which made it impossible to log in through Remote Desktop Gateway (RDG) using two-factor authentication.
- Fixed an error in NPS Radius Extension where the timeout for the second authentication factor request was always different.

Appendix

Event log

The list of Axidian Access Server and Axidian Key Server events

Roles

Access rights of the administrator, the operator, and the inspector

Event log

This section contains the lists of Axidian Access Server and Axidian Key Server events.

Axidian Access Server events

Axidian Access records events of the following types:

- Information (an informational message, no problems or errors)
- Error (problems or errors that require user intervention)
- Warning (problems or errors are possible)

Information

Code	Message text	Event attributes
1000	The user has been authenticated with the provided authenticator.	Application, User, Computer, Authentication method, Authenticator comment.
1001	The Axidian Access Server has been started.	Computer.
1002	The Axidian Access Server has been stopped.	Computer.
1003	The user has been identified by the provided authentication method.	Application, User, Computer, Authentication method.
1004	An authenticator for the target user has been registered by the administrator.	Application, Target user, Administrator, Computer, Authentication method, Authenticator comment.
1005	The 'Maximum number of authenticators' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1006	The 'Right to register an authenticator' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.

Code	Message text	Event attributes
1007	The 'Right to modify an authenticator' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1008	The 'Right to delete an authenticator' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1009	The 'Right to change the authenticator comment during registration' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1010	The 'Right to change the authenticator comment' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1011	The 'Cache data starting from' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1012	The 'Cache data up to' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1014	The 'Cache data during the period' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1015	The 'User substitution start date' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1016	The 'User substitution end date' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1017	The 'List of substitutes' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1018	The 'Allow caching' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.

Code	Message text	Event attributes
1019	The 'Allow substitution' setting for the target user has been changed by the administrator.	Administrator, Computer, Target user, Old value, New value.
1020	A license has been issued to the target directory object by the administrator.	Computer, Administrator, Application ID, Target directory object, License type.
1021	A license has been added by the user.	License ID, Start date, End date, Quantity, Description, User, Computer.
1022	A license has been deleted by the user.	License ID, Start date, End date, Quantity, Description, User, Computer.
1023	The license of the target directory object has been deleted by the administrator.	License ID, Computer, Administrator, Application ID, Target directory object, License type.
1024	The authenticator of the target user has been modified by the administrator.	Application, Target user, Administrator, Computer, Authentication method, Authenticator comment.
1025	The authenticator comment of the target user has been changed by the administrator.	Application, Target user, Administrator, Computer, Authentication method, Old authenticator comment, New authenticator comment.
1026	The authenticator of the target user has been deleted by the administrator.	Application, Target user, Administrator, Computer, Authentication method, Authenticator comment.
1027	The license of the target user has been deleted by the server.	License ID, Application ID, Target user.
1028	The target directory object has been added by the administrator to the access group for the target application.	Application, Initiator, Computer, Target application, Target directory object, Policy, Access group, Scope.
1029	The target directory object has been removed by the administrator from the access group for the target application.	Application, Initiator, Computer, Target application, Target directory object, Policy, Access group, Scope.
1030	A device has been added by the user.	Application, User, Computer, Device, Comment, Serial number.

Code	Message text	Event attributes
1031	A device has been deleted by the user.	Application, User, Computer, Device, Comment, Serial number, Device ID.
1032	An authentication method for the target user has been forbidden by the administrator.	Application, Administrator, Computer, Target user, Authentication method.
1033	An authentication method for the target user has been allowed by the administrator.	Application, Administrator, Computer, Target user, Authentication method.
1034	The authenticator of the target user has been blocked by the administrator.	Application, Administrator, Computer, Target user, Authentication method, Authenticator comment.
1035	The authenticator of the target user has been unblocked by the administrator.	Application, Administrator, Computer, Target user, Authentication method, Authenticator comment.
1036	An authentication method has been blocked by the user.	Application, User, Computer, Authentication method.
1037	An authentication method for the target user has been unblocked by the administrator.	Application, Administrator, Computer, Target user, Authentication method.
1038	An SMS message has been sent by the user.	Application, Initiator, Computer, Authentication method, Phone, Sending time, Sending parameters, Message ID.
1039	An email message has been sent by the user.	Application, User, Computer, Authentication method, Email.
1040	Some of the server catalogs are unavailable.	Computer, Error description.
1041	An authenticator has been modified by the user.	Application, User, Computer, Authentication method, Authenticator comment.
1042	An authenticator has been registered by the user.	Application, User, Computer, Authentication method, Authenticator comment.
1043	An authenticator comment has been changed by the user.	Application, User, Computer, Authentication method, Old authenticator comment, New authenticator comment.

Code	Message text	Event attributes
1044	An authenticator has been deleted by the user.	Application, User, Computer, Authentication method, Authenticator comment.
1045	The authenticator of the target user has been disabled by the administrator.	Application, Administrator, Computer, Target user, Authentication method, Authenticator comment.
1046	The authenticator of the target user has been enabled by the administrator.	Application, Administrator, Computer, Target user, Authentication method, Authenticator comment.
1047	The license of the target directory object has been deleted by the administrator.	Computer, Administrator, Application ID, Target directory object, License type.
1048	A license has been added by the user.	License ID, Start date, End date, Quantity, Description, User, Computer, Application ID.
1049	A license has been deleted by the user.	License ID, Start date, End date, Quantity, Description, User, Computer, Application ID.
1050	The license of the target user has been deleted by the server.	Application ID, Target user.
1051	The policy has been created successfully.	ID, Name, Description, Priority, New directory objects, New applications, Application, Initiator, Computer.
1052	The policy has been deleted.	ID, Name, Description, Priority, Deleted directory objects, Deleted applications, Application, Initiator, Computer.
1053	The policy has been modified.	ID, Name, Description, Priority, New directory objects, New applications, Deleted directory objects, Deleted applications, Application, Initiator, Computer.
1054	The user password has been synchronized successfully.	Initiator, Computer, Client user, Client computer.
1055	The user has signed in successfully with the provided password.	Initiator, Computer, Client user, Client computer.

Code	Message text	Event attributes
1056	The user has signed in successfully with a cached authenticator.	Initiator, Sign-in method, Authenticator comment, Computer, Client user, Client computer.
1057	The user has signed in successfully with the provided authenticator.	Initiator, Sign-in method, Authenticator comment, Computer, Client user, Client computer.
1058	The user has removed the sign-in device. The session has been terminated.	Initiator, Authenticator comment, Computer, Client user, Client computer.
1059	The user has signed out.	Initiator, Authenticator comment, Computer, Client user, Client computer.
1060	The user has removed the sign-in device. The computer has been locked.	Initiator, Authenticator comment, Computer, Client user, Client computer.
1061	The user has locked the computer.	Initiator, Authenticator comment, Computer, Client user, Client computer.
1062	The user has unlocked the computer with the provided password.	Initiator, Computer, Client user, Client computer.
1063	The user has unlocked the computer with a cached authenticator.	Initiator, Sign-in method, Authenticator comment, Computer, Client user, Client computer.
1064	The user has unlocked the computer with the provided authenticator.	Initiator, Sign-in method, Authenticator comment, Computer, Client user, Client computer.
1065	The user has reconnected to their terminal session.	Initiator, Authenticator comment, Computer, Client user, Client computer.
1067	The user password has been changed automatically.	Initiator, Computer, Client user, Client computer.
1068	The user password has been changed successfully.	Initiator, Computer, Client user, Client computer.
1069	The user password is out of sync.	Initiator, Computer, Client user, Client computer.
1070	The user has been authenticated for the target application.	Application, Initiator, Computer, Target user.

Code	Message text	Event attributes
1073	Authentication method settings have been changed:	Application, Initiator, Computer, Provider, Settings.
1076	Authenticator rights settings have been changed — Success:	Policy, User, Registration, Re-registration, Deletion, Change the comment during registration only, Change the comment, Application, Initiator, Computer.
1077	An account setting has been created.	Policy, Target user, Role, Linked account, Application, Name, Password is set, Password is random, Account is disabled, Require authentication at sign-in, Use OS credentials, Name conversion format, Description, Application, Initiator, Computer.
1078	An account setting has been deleted.	Policy, Target user, Role, Linked account, Application, Name, Password is set, Password is random, Account is disabled, Require authentication at sign-in, Use OS credentials, Name conversion format, Description, Application, Initiator, Computer.
1079	An account setting has been changed.	Policy, Target user, Role, Linked account, Application, Name, Password is set, Password is random, Account is disabled, Require authentication at sign-in, Use OS credentials, Name conversion format, Description, Application, Initiator, Computer.
1082	Authentication method settings for the user have been changed:	Target user, Provider, Application, Initiator, Computer, Settings.
1085	Authentication method settings for the module have been changed:	Module, Provider, Policy, Policy ID, Application, Initiator, Computer, Settings.
1086	The user data caching settings have been changed successfully:	Target application, Target user, Old values, New values, Application, Initiator, Computer.
1087	The user data caching settings in the policy have been changed successfully:	Target application, Policy, Old values, New values, Application, Initiator, Computer.

Code	Message text	Event attributes
1088	The phone number has been registered successfully.	User, Initiator, API user, Phone.
1089	The phone number has been changed successfully.	User, Initiator, API user, Phone.
1090	The phone number has been deleted successfully.	User, Initiator, API user.
1091	The user password has been changed automatically.	Target user, Application, Initiator, Computer.
1092	The user password has been changed successfully.	Target user, Application, Initiator, Computer.
1093	The authentication method policy has been changed:	Provider, Policy, Policy ID, Application, Initiator, Computer, Settings.
1094	The license revocation service has finished. Licenses have been revoked because a user, a policy, or a policy application was deleted, or because a user was excluded from the policy under which the license was issued.	Licenses revoked, Failed to revoke.
1095	The license of the target user has been deleted by the server because the user inactivity period has been exceeded.	Application ID, Target user.
1096	The license revocation service has finished. Licenses have been revoked because of user inactivity.	Licenses revoked, Failed to revoke.
1097	The Axidian ESSO Agent has been initialized successfully and started.	Initiator, Computer, Process ID, Computer.
1098	The Axidian ESSO Agent is shutting down.	Initiator, Computer, Process ID, Computer.
1099	The application has been terminated forcibly.	Initiator, Computer, Application description, Application executable path, Computer, Command line.

Code	Message text	Event attributes
1100	The application form has been closed forcibly.	Initiator, Computer, Application description, Application executable path, Form type, Computer, Command line.
1101	The user has authenticated in Axidian ESSO.	Initiator, Computer, User name, Computer.
1102	The user has re-authenticated in Axidian ESSO.	Initiator, Computer, User name, Computer.
1103	The user has signed in to the application.	Initiator, Computer, Application description, Application executable path, User name, User account name in the application, Computer, SSO account description, Command line.
1104	The user password has been changed successfully.	Initiator, Computer, Application description, Application executable path, User name, User account name in the application, Computer, User description, Command line.
1105	A network connection has been added for the Axidian ESSO user.	Initiator, Computer, User name, Connection drive, Connection path, Computer.
1106	A network connection has been removed for the Axidian ESSO user.	Initiator, Computer, User name, Connection drive, Computer.
1107	An error occurred while removing the network connection for the Axidian ESSO user.	Initiator, Computer, User name, Connection drive, Computer.
1108	The user has unlocked the application successfully.	Initiator, Computer, Application description, Application executable path, User name, User account name in the application, Computer, User description, Command line.
1109	The substitute user has authenticated in Axidian ESSO.	Initiator, Computer, User name, Substitute user, Computer.
1110	The substitute user has re-authenticated in Axidian ESSO.	Initiator, Computer, User name, Substitute user, Computer.

Code	Message text	Event attributes
1111	The user password has been generated and changed automatically.	Initiator, Computer, Application description, Application executable path, User name, User account name in the application, Computer, User description, Command line.
1112	A new application has been added.	Name, Description, Integration module, Application, Initiator, Computer.
1113	The application has been deleted.	Name, Description, Integration module, Application, Initiator, Computer.
1114	The application parameters have been changed.	Application name, Description, List of changes, Module, Initiator, Computer.
1115	A Telegram message has been sent to the user.	Integration module, Initiator, Computer, Authentication method.
1116	The Telegram contact has been registered successfully.	User, Initiator, Phone.
1117	The user has been authenticated for the purpose of creating a simple electronic signature.	Signature verification key, Business application name, Device ID, API user, User.
1118	Sending messages to users has been resumed.	Computer, Authentication method.
1119	The user has been authenticated on the device.	Provided authenticator, Operation key, Business application name, Device ID, Authentication method, API user.
1120	The user has been authenticated to sign an electronic document.	Provided identifier, Signature verification key, Business application name, Electronic document ID, Authentication method, API user.
1121	The authenticator registration process has been initiated.	Application, Initiator, Computer, Authentication method.
1122	User authentication has been initiated.	Integration module, Initiator, Computer, Authentication method, Application.

Code	Message text	Event attributes
1123	The user has removed the mobile device from Exchange quarantine successfully.	Initiator, Device ID.

Error

Code	Message text	Event attributes	Causes
2000	An error occurred while trying to authenticate the user by the provided authentication method.	Application, User, Computer, Authentication method, Error description.	Invalid authenticator. An authenticator is considered invalid if: An authenticator of the specified type has not been registered for the user. The authenticator has not been found in the list of the user authenticators. The data storage is corrupted.
2001	An error occurred while trying to start the Axidian Access Server.	Computer, Error description.	An attempt to start the Axidian Access Server manually has failed. Possible causes: An error in the server configuration file. The server is already running. The data storage is corrupted. An IIS error.
2002	An error occurred while trying to stop the Axidian Access Server.	Computer, Error description.	An attempt to stop the Axidian Access Server manually has failed. Possible causes: Insufficient rights to perform the operation. The Axidian Access Server is already stopped.
2003	An error occurred while trying to identify the user by the provided authentication method.	Application, Computer, Authentication method, Error description.	An attempt to identify the user by the selected authentication method has failed. Possible causes: The selected authentication method is disabled or blocked for the user. The authenticator has not been found in the list of the user authenticators. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2004	An error occurred while the administrator was trying to register an authenticator for the target user.	Application, Target user, Administrator, Computer, Authentication method, Authenticator comment, Error description.	An attempt by the administrator to register an authenticator has failed. Possible causes: The authenticator provider has been deleted. The data storage is corrupted. No rights to manage authenticators.
2005	An error occurred while the administrator was trying to modify the authenticator of the target user.	Application, Target user, Administrator, Computer, Authentication method, Old authenticator comment, New authenticator comment, Error description.	An attempt by the administrator to modify an authenticator has failed. Possible causes: The authenticator provider has been deleted. The data storage is corrupted. No rights to modify the authenticator.
2006	An error occurred while the administrator was trying to delete the authenticator of the target user.	Application, Target user, Administrator, Computer, Authentication method, Authenticator comment, Error description.	An attempt by the Axidian Access administrator to delete a user authenticator has failed. Possible causes: The authenticator has already been deleted. The data storage is corrupted.
2007	An error occurred while the administrator was trying to issue a license for the target directory object.	Computer, Administrator, Application ID, Target directory object, License type, Error description.	An attempt by the Axidian Access administrator to issue a license for the target directory object has failed. Possible causes: An error in the license file. The Axidian Access Server is inactive. The data storage is corrupted.
2008	An error occurred while the administrator was trying to delete the license of the target directory object.	Computer, Administrator, Application ID, Target directory object, Error description.	An attempt by the Axidian Access administrator to delete the license of the target directory object has failed. Possible causes: The license has already been deleted. The Axidian Access Server is inactive. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2009	An error occurred while the user was trying to delete a license.	License ID, User, Computer, Error description.	An attempt by the Axidian Access user to delete a license has failed. Possible causes: The license has already been deleted. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2010	An error occurred while the user was trying to add a license.	License ID, User, Computer, Error description.	An attempt by the Axidian Access user to add a license has failed. Possible causes: An error in the license file. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2011	An error occurred while the server was trying to delete the license of the target user.	License ID, Application ID, Target user, Error description.	An attempt by the server to delete the license of the target user has failed. Possible causes: The license has already been deleted. The data storage is corrupted.
2012	An error occurred while the administrator was trying to change the 'Maximum number of authenticators' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Maximum number of authenticators" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2013	An error occurred while the administrator was trying to change the 'Right to register an authenticator' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Right to register an authenticator" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2014	An error occurred while the administrator was trying to change the 'Right to modify an authenticator' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Right to modify an authenticator" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2015	An error occurred while the administrator was trying to change the 'Right to delete an authenticator' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Right to delete an authenticator" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2016	An error occurred while the administrator was trying to change the 'Right to change the authenticator comment during registration' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Right to change the authenticator comment during registration" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2017	An error occurred while the administrator was trying to change the 'Right to change the authenticator comment' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Right to change the authenticator comment" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2018	An error occurred while the administrator was trying to change the 'Cache data starting from' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Cache data starting from" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2019	An error occurred while the administrator was trying to change the 'Cache data up to' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Cache data up to" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2020	An error occurred while the administrator was trying to change the 'Cache data during the period' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Cache data during the period" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2021	An error occurred while the administrator was trying to change the 'User substitution start date' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "User substitution start date" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2022	An error occurred while the administrator was trying to change the 'User substitution end date' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "User substitution end date" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2023	An error occurred while the administrator was trying to change the 'List of substitutes' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "List of substitutes" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2024	An error occurred while the administrator was trying to change the 'Allow caching' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Allow caching" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2025	An error occurred while the administrator was trying to change the 'Allow substitution' setting for the target user.	Administrator, Computer, Target user, Old value, New value, Error description.	An attempt by the system administrator to change the "Allow substitution" parameter for the user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2026	An error occurred while the administrator was trying to add the target directory object to the access group for the target application.	Application, Initiator, Computer, Target application, Target directory object, Policy, Access group, Scope, Error description.	An attempt to add the target directory object to the access group for the target application has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2027	An error occurred while the administrator was trying to remove the target directory object from the access group for the target application.	Application, Initiator, Computer, Target application, Target directory object, Policy, Access group, Scope, Error description.	An attempt to remove the target directory object from the access group for the target application has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2028	An error occurred while the user was trying to add a device.	Application, User, Computer, Device, Comment, Serial number, Error description.	An attempt by the system user to add a device has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2029	An error occurred while the user was trying to delete a device.	Application, User, Computer, Device, Comment, Serial number, Device ID, Error description.	An attempt by the system user to delete a device has failed. Possible causes: The device has already been deleted. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2030	An error occurred while the administrator was trying to forbid an authentication method for the target user.	Application, Administrator, Computer, Target user, Authentication method, Error description.	An attempt by the system administrator to forbid an authentication method for the target user has failed. Possible causes: This method is already forbidden. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2031	An error occurred while the administrator was trying to allow an authentication method for the target user.	Application, Administrator, Computer, Target user, Authentication method, Error description.	An attempt by the system administrator to allow an authentication method for the target user has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2032	An error occurred while the administrator was trying to block the authenticator of the target user.	Application, Administrator, Computer, Target directory object, Authentication method, Authenticator comment, Error description.	An attempt by the system administrator to block the authenticator of the target user has failed. Possible causes: The authenticator is already blocked. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2033	An error occurred while the administrator was trying to unblock the authenticator of the target user.	Application, Administrator, Computer, Target user, Authentication method, Authenticator comment, Error description.	An attempt by the system administrator to unblock the authenticator of the target user has failed. Possible causes: The authenticator is already blocked. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2034	An error occurred while the user was trying to block an authentication method.	Application, User, Computer, Authentication method, Error description.	An attempt by the user to block an authentication method has failed. Possible causes: The authentication method is already blocked. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2035	An error occurred while the administrator was trying to unblock an authentication method for the target user.	Application, Administrator, Computer, Target user, Authentication method, Error description.	An attempt by the user to unblock an authentication method has failed. Possible causes: The authentication method is already unblocked. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2036	An error occurred while trying to send an SMS message.	Application, Initiator, Computer, Authentication method, Phone, Error description, Sending time, Sending parameters, Message ID.	An attempt to send an SMS has failed. Possible causes: An error occurred while connecting to the SMPP server. The phone number is not specified.
2037	An error occurred while the user was trying to send an email message.	Application, User, Computer, Authentication method, Email.	An attempt to send an email message has failed. Possible causes: An error occurred while connecting to the SMTP server. The user email is not specified.
2038	The server storage is unavailable.	Computer, Error description.	An attempt to connect to the Axidian Access Server has failed. Possible causes: The Axidian Access Server is inactive. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2039	The server catalogs are unavailable.	Computer, Error description.	
2040	The server has not been started.	Computer, Error description.	An attempt to start the Axidian Access Server has failed. Possible causes: The Axidian Access Server is already running. An error in the configuration file. The data storage is corrupted.
2041	An error occurred while the user was trying to modify an authenticator.	Application, User, Computer, Authentication method, Old authenticator comment, New authenticator comment, Error description.	An attempt by the user to modify an authenticator has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2042	An error occurred while the user was trying to register an authenticator.	Application, User, Computer, Authentication method, Authenticator comment, Error description.	An attempt by the user to register an authenticator has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2043	An error occurred while the user was trying to delete an authenticator.	Application, User, Computer, Authentication method, Authenticator comment, Error description.	An attempt by the user to delete an authenticator has failed. Possible causes: The authenticator has already been deleted. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2044	An error occurred while the administrator was trying to disable the authenticator of the target user.	Application, Administrator, Computer, Target directory object, Authentication method, Authenticator comment, Error description.	An attempt by the system administrator to disable the authenticator of the target user has failed. Possible causes: The authenticator is already disabled. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2045	An error occurred while the administrator was trying to enable the authenticator of the target user.	Application, Administrator, Computer, Target user, Authentication method, Authenticator comment, Error description.	An attempt by the system administrator to enable the authenticator of the target user has failed. Possible causes: The authenticator is already enabled. Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2046	An error occurred while the user was trying to add a license.	License ID, User, Computer, Application ID, Error description.	An attempt by the user to add a license has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted. An error in the license file.
2047	An error occurred while the user was trying to delete a license.	License ID, User, Computer, Application ID, Error description.	An attempt by the user to delete a license has failed. Possible causes: Insufficient rights. The Axidian Access Server is inactive. The data storage is corrupted.
2048	An error occurred while the server was trying to delete the license of the target user.	Application ID, Target user, Error description.	An attempt by the server to delete a license has failed.
2049	An error occurred while creating a policy.	ID, Name, Description, Priority, New directory objects, New applications, Error, Application, Initiator, Computer.	The policy has not been created.
2050	An error occurred while deleting a policy.	ID, Name, Description, Priority, Deleted directory objects, Deleted applications, Error, Application, Initiator, Computer.	The policy has not been deleted.

Code	Message text	Event attributes	Causes
2051	An error occurred while modifying a policy.	ID, Name, Description, Priority, New directory objects, New applications, Deleted directory objects, Deleted applications, Error, Application, Initiator, Computer.	The policy has not been modified.
2052	The Axidian Access Server has not been found.	Computer, Error description, Error code.	An error occurred while trying to connect to the Axidian Access Server. An incorrect URL is specified. A DNS problem.
2053	Failed to change the user password.	Initiator, Computer, Client user, Client computer, Error description, Error code.	The user password has not been changed.
2054	Failed to change the user password automatically.	Initiator, Computer, Client user, Client computer, Error description, Error code.	The user password has not been changed automatically.
2055	Failed to synchronize the user password.	Initiator, Computer, Client user, Client computer, Error description, Error code.	An error occurred while synchronizing the domain user password with the password in the Axidian Access storage. Possible causes: The Axidian Access Server is unavailable. The data storage is corrupted.
2058	Authentication method settings have been changed:	Application, Initiator, Computer, Provider, Settings, Error description.	The authentication method settings have not been changed.
2061	Authenticator rights settings have been changed — Failure:	Policy, User, Registration, Re-registration, Deletion, Change the comment during registration only, Change the comment, Application, Initiator, Computer, Error.	The authenticator rights settings have not been changed.

Code	Message text	Event attributes	Causes
2062	An error occurred while creating an account setting.	Policy, Target user, Role, Linked account, Application, Name, Password is set, Password is random, Account is disabled, Require authentication at sign-in, Use OS credentials, Name conversion format, Description, Application, Initiator, Computer, Error.	The account setting has not been created.
2063	An error occurred while deleting an account setting.	Policy, Target user, Role, Linked account, Application, Name, Password is set, Password is random, Account is disabled, Require authentication at sign-in, Use OS credentials, Name conversion format, Description, Application, Initiator, Computer, Error.	The account setting has not been deleted.
2064	An error occurred while changing an account setting.	Policy, Target user, Role, Linked account, Application, Name, Password is set, Password is random, Account is disabled, Require authentication at sign-in, Use OS credentials, Name conversion format, Description, Application, Initiator, Computer, Error.	The account setting has not been changed.
2067	Authentication method settings for the user have been changed:	Target user, Provider, Application, Initiator, Computer, Settings, Error description.	The authentication method settings for the user have not been changed.

Code	Message text	Event attributes	Causes
2070	Authentication method settings for the module have been changed:	Module, Provider, Policy, Policy ID, Application, Initiator, Computer, Settings, Error description.	The authentication method settings for the module have not been changed.
2071	An error occurred while trying to change the user data caching settings:	Target application, Target user, Old values, New values, Application, Initiator, Computer, Error description.	An attempt to change the user data caching settings has failed. Possible causes: Insufficient rights to perform the operation. The data storage is corrupted.
2072	An error occurred while trying to change the user data caching settings in the policy:	Target application, Policy, Old values, New values, Application, Initiator, Computer, Error description.	An attempt to change the user data caching settings in the policy has failed. Possible causes: Insufficient rights to perform the operation. The data storage is corrupted.
2073	An error occurred while registering a phone number.	User, Initiator, API user, Error description, Error initiator, Phone.	An attempt to register a phone number has failed. Possible causes: Insufficient rights to perform the operation. The EAPhoneServer component is unavailable. A non-existent user is specified. The data storage is corrupted.
2074	An error occurred while changing a phone number.	User, Initiator, API user, Error description, Error initiator, Phone.	An attempt to change a phone number has failed. Possible causes: Insufficient rights to perform the operation. The EAPhoneServer component is unavailable. A non-existent user is specified. The data storage is corrupted.
2075	An error occurred while deleting a phone number.	User, Initiator, API user, Error description, Error initiator.	An attempt to delete a phone number has failed. Possible causes: Insufficient rights to perform the operation. The EAPhoneServer component is unavailable. A non-existent user is specified. The data storage is corrupted.

Code	Message text	Event attributes	Causes
2076	Failed to change the user password automatically.	Target user, Application, Initiator, Computer, Error description.	An attempt to change the password to a random one has failed. Possible causes: Insufficient rights to perform the operation. The Axidian Access Server installation is corrupted. The password change request has been blocked by security policies.
2077	Failed to change the user password.	Target user, Application, Initiator, Computer, Error description.	An attempt by the user to change the password manually has failed (at the first sign-in, in the Windows Security window after pressing [Ctrl]+[Alt]+[Del]).
2078	An error occurred while changing the authentication method policy:	Provider, Policy, Policy ID, Application, Initiator, Computer, Settings, Error description.	The authentication method policy has not been changed.
2079	An error occurred while saving the user password to the Axidian ESSO database.	Initiator, Computer, Application description, Application executable path, User name, User account name in the application, Computer, Error description, Error code, Command line.	There is no connection to the Axidian Access Server or to the domain controller.
2080	An error occurred while loading the ESSO data for the authenticated user.	Initiator, Computer, User name, Computer, Error description, Error code.	The Axidian Enterprise Single Sign-On user data in Active Directory is corrupted. Failed to close the application form forcibly.
2081	A failure has been detected in the Axidian ESSO Agent.	Initiator, Computer, Computer, Error description, Error code.	Failed to terminate the application process.

Code	Message text	Event attributes	Causes
2082	An error occurred while adding a network connection for the user.	Initiator, Computer, User name, Connection drive, Connection path, Computer, Error description, Error code.	The network drive is full. There is no connection to the Axidian Access Server.
2083	An error occurred while adding a new application.	Name, Description, Integration module, Application, Initiator, Computer, Error description.	An attempt to add a new application has failed: Insufficient rights to perform the operation. There is no connection to the Axidian Access Server.
2084	An error occurred while deleting an application.	Name, Description, Integration module, Application, Initiator, Computer, Error description.	An attempt to delete an application has failed: Insufficient rights to perform the operation. There is no connection to the Axidian Access Server.
2085	An error occurred while changing the application parameters.	Name, Description, List of changes, Application, Initiator, Computer, Error description.	An attempt to change the application parameters has failed: Insufficient rights to perform the operation. There is no connection to the Axidian Access Server.
2086	Failed to deliver the message to the central log server.	Application, Initiator, Computer, Type, Category, Message ID, Description.	An attempt to send the message to the central log server has failed: There is no connection to the Axidian Access Log Server.
2087	An error occurred while trying to send a Telegram message.	Application, Initiator, Authentication method, Error description.	One or more errors occurred. An error occurred while sending the request: there is no access or connection from the Core Server to api.telegram.org.

Code	Message text	Event attributes	Causes
2088	An error occurred while registering a Telegram contact.	User, Initiator, Error description, Phone.	Access denied. The user cannot be authorized. One or more errors occurred. The user does not have the following rights: ReadUserAuthenticators for the Authenticator Management application: the Telegram service account has not been added to the global administrators of the Management Console.
2089	An error occurred while trying to authenticate the user for the purpose of creating a simple electronic signature.	Error description, Business application name, API user, Device ID.	An error occurred while authenticating the user when creating a simple electronic signature.
2090	A potential spam attack has been detected. Sending messages has been suspended.	Computer, Authentication method, Time interval start, Time interval end, Number of sendings, Number of successful sign-ins, Percentage of successful sign-ins.	An attempt to send a message has failed because suspicious activity was detected for this authentication method.
2091	An error occurred while trying to authenticate the user on the device.	Error description, Provided identifier, Operation key, Business application name, Device ID, Authentication method, API user.	The code entered on the device is incorrect.
2092	An error occurred while trying to authenticate the user to sign an electronic document.	Error description, Provided identifier, Signature verification key, Business application name, Electronic document ID, Authentication method, API user.	The sign-in attempt has failed. Possible causes: An incorrect login. An incorrect password.

Code	Message text	Event attributes	Causes
2093	The user entered an OTP whose length does not match the authentication provider settings.	Authentication method, Target user, Value at registration, Current value.	The actual length of the entered OTP does not match the OTP length set in the provider.
2094	The OTP refresh period in the user authenticator does not match the authentication provider settings.	Authentication method, Target user, Value at registration, Current value.	The actual OTP refresh period does not match the refresh period that has been set.
2095	The OTP algorithm in the user authenticator does not match the authentication provider settings.	Authentication method, Target user, Value at registration, Current value.	The actual OTP algorithm does not match the algorithm that has been set.
2096	Failed to remove the mobile device from Exchange quarantine.	Initiator, Device ID.	An attempt to remove the device from quarantine has failed.
2097	Failed to block the mobile device on the Exchange server.	Initiator, Device ID.	An attempt to block the mobile device has failed.

Warning

Code	Message text	Event attributes	Causes
3000	The OTP comparison window in the user authenticator does not match the authentication provider settings.	Authentication method, Target user, Value at registration, Current value.	The actual OTP comparison window does not match the comparison window that has been set.

Code	Message text	Event attributes	Causes
3001	The OTP synchronization window in the user authenticator does not match the authentication provider settings.	Authentication method, Target user, Value at registration, Current value.	The actual OTP synchronization window does not match the synchronization window that has been set.
3002	The user has blocked the mobile device on the Exchange server.	Initiator, Device ID.	The attempt to block the device has succeeded.

Axidian Key Server events

> **NOTE**
> You can view the event log through the Windows Event Log (if the Windows storage is used) or with third-party tools through SQL (if the SQL storage is used).

Code	Category	Service	Message text
1000	Information	Axidian Key	The authentication request for the user {userLogin} has been sent. Application: {appId}
1001	Information	Axidian Key	The registration request for the user {userLogin} has been sent. Application: {appId}
1002	Information	Axidian Key	The response of the user {userLogin} to the authentication request has been processed. Application: {appId}. Result: {state}. Message: {message}
1003	Information	Axidian Key	The certificate installation request for the user {userLogin} has been sent. Application: {appId}
1004	Information	Axidian Key	The registration of the user {userLogin} has been completed successfully. Application: {appId}
1005	Information	Axidian Key	The deletion of the user {userLogin} has been completed successfully. Application: {appId}
2000	Error	Axidian Key	An error occurred while sending the authentication request for the user {userLogin}. Application: {appId}. Error:

Code	Category	Service	Message text
2001	Error	Axidian Key	An error occurred while sending the registration request for the user {userLogin}. Application: {appId}. Error:
2002	Error	Axidian Key	An error occurred while sending the certificate installation request for the user {userLogin}. Application: {appId}. Error:
2003	Error	Axidian Key	An error occurred while registering the user {userLogin}. Application: {appId}. Error:
2004	Error	Axidian Key	An error occurred while deleting the user {userLogin}. Application: {appId}. Error:

Roles

This section contains the tables of roles and rights of the following users:

- **Administrator** — full access to all system functions and settings
- **Operator** — rights to change user settings and authentication device settings only
- **Inspector** — read access to all system data and settings

Registering authenticators

Operation	Administrator	Operator	Inspector
Register the first authenticator	✓	✓	✗
Register any authenticator	✓	✓	✗

User authenticators

Operation	Administrator	Operator	Inspector
View user authenticators	✓	✓	✓
Disable user authenticators	✓	✓	✗
Enable user authenticators	✓	✓	✗
Delete user authenticators	✓	✓	✗
Unblock user authentication methods	✓	✓	✗
Get user authentication methods	✓	✓	✓
Enable user authentication methods	✓	✓	✗

Operation	Administrator	Operator	Inspector
Disable user authentication methods	✓	✓	✗
View the state of a user authentication method	✓	✓	✓
Change authentication method settings	✓	✗	✗
View authentication method settings	✓	✗	✓
Change user authentication method settings	✓	✓	✗
View user authentication method settings	✓	✓	✓
Change application authentication method settings	✓	✓	✗
View application authentication method settings	✓	✓	✓
Change access rights for an authenticator	✓	✓	✗
View access rights for an authenticator	✓	✓	✓
Change authentication method settings in a policy	✓	✓	✗
View authentication method settings in a policy	✓	✓	✓

Licenses

Operation	Administrator	Operator	Inspector
Register licenses	✓	✗	✗
Unregister licenses	✓	✗	✗

Operation	Administrator	Operator	Inspector
View registered licenses	✓	✓	✓
Issue a license to a user	✓	✓	✗
Revoke a license from a user	✓	✓	✗
View user licenses	✓	✓	✓

Administrative access

Operation	Administrator	Operator	Inspector
Add a directory object to an access group	✓	✗	✗
Remove a directory object from an access group	✓	✗	✗
View access group members	✓	✓	✓

Authentication devices

Operation	Administrator	Operator	Inspector
Register a device	✓	✗	✗
Update a device	✓	✗	✗
View the device list	✓	✓	✓
Delete a device	✓	✗	✗

Policies

Operation	Administrator	Operator	Inspector
Create a policy	✓	✗	✗
Modify a policy	✓	✗	✗
Delete a policy	✓	✗	✗
View the policy list	✓	✓	✓
View the list of policy users	✓	✓	✓
Modify the list of policy users	✓	✓	✗
View the list of policy applications	✓	✓	✗
Modify the list of policy applications	✓	✗	✗
View the policy priority	✓	✓	✓
Change the policy priority	✓	✗	✗

Caching user data on a computer

Operation	Administrator	Operator	Inspector
Enable caching	✓	✓	✗
Disable caching	✓	✓	✗
Check the cache state	✓	✓	✗

Operation	Administrator	Operator	Inspector
Change caching settings	✓	✗	✗
View caching settings	✓	✓	✓

ESSO accounts

Operation	Administrator	Operator	Inspector
Add an account	✓	✗	✗
Delete an account	✓	✗	✗
View user accounts	✓	✓	✓
Read the state of user accounts	✓	✓	✗
Change the password of user accounts	✓	✗	✗
View the password state of a user account	✓	✓	✓
Change the name of a user account	✓	✓	✗
View the name of a user account	✓	✓	✓
Change the settings of a user account	✓	✓	✗
View the settings of a user account	✓	✓	✓
Change the additional data of a user account	✓	✓	✗
View the additional data of a user account	✓	✓	✓

Operation	Administrator	Operator	Inspector
Set a random password for a user account	✓	✗	✗
Change the account password	✓	✗	✗

Applications

Operation	Administrator	Operator	Inspector
View the list of business applications	✓	✓	✓
Create a business application	✓	✗	✗
Modify a business application	✓	✓	✗
Delete a business application	✓	✗	✗