

ACTIVE DIRECTORY PASSWORD AGENT INSTALLATION AND TROUBLESHOOTING MANUAL

Inalogy a.s.
Černyševského 48, 851 01 Bratislava
55043712. SK212185377

+421 2 3810 1152
info@inalogy.com
www.inalogy.com

Table of Contents

1. Introduction	3
2. Installation.....	4
2.1 Installation prerequisites	4
2.1.1 .NET framework	4
2.1.2 Communication between AD DC and midPoint	4
2.1.3 User in midPoint	4
2.1.4 midpoint SSL TRUST configuration	5
2.1.5 User in Active Directory	5
2.1.6 Service client certificate generation	6
2.2 Installer installation	7
3. Recommended postinstall activities.....	10
3.1 DC controller firewall configuration modification	10
3.2 Elimination of Password change cycling	12
3.3 Password agent authorization	13
3.4 Limiting password agent MS AD user object rights	14
4. Testing / Troubleshooting	17
4.1 AD Basic test	17
4.2 Troubleshooting installation with installer	19
4.3 ADPasswordAgent troubleshooting	19
4.4 Location of registry password hive	20
4.5 Location of file storage.....	20
4.6 Location of server certificate	20
4.7 midPointUpdatingService Logging troubleshooting.....	21

1. Introduction

This document describes installation, troubleshooting and building of midPoint password sync agent. Used components:

- **ADPasswordFilter.dll** runs in the context of an AD Domain Controller and listens for AD password change requests.
- **MidPointUpdatingService.exe** is installed and registered as a Microsoft Windows OS service. It continuously checks for presence of an ActionCall in registry hive and optionally the local file storage. If any ActionCall is present, it is executed against the configured midPoint instance. In case it's not successful due to recoverable error (e.g., Network Connection error), two actions are executed. Passwords are moved from secured registry hive to local file storage in encrypted form and after a couple of attempts to retry the transfer of password change in a rising time delay. When non-recoverable error occurs, defined by consuming defined maximum retry, the ActionCall is dequeued and released. The information is written to the log.

2. Installation

This chapter defines installation steps and installation prerequisites for successful application service installation.

2.1 Installation prerequisites

- minimum required .NET 4.7.2
- communication between AD and midPoint is enabled
- user object in midPoint (e.g., called “midpoint-agent”) - to receive password changes into midpoint via REST API
 - user object authorization to change the password of other midPoint user objects
 - midPoint SSL configuration
- user in AD (e.g., called “midpoint”) - to register and run the midpoint service (midpointUpdatingService)
 - Service client certificate generation
- domain admin rights to run the Installer and register the service
- trust relationship to the certification authority used to sign midPoint HTTPS interface certificate.

2.1.1 .NET framework

To verify already installed version, please see Microsoft manual:

<https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/how-to-determine-which-versions-are-installed#detect-net-framework-45-and-later-versions>

If you don't have already installed .NET version 4.7.2 or 4.8.x, please follow Microsoft installation guide:

<https://docs.microsoft.com/en-us/dotnet/framework/install/>

2.1.2 Communication between AD DC and midPoint

To verify, whether TCP communication is already opened from AD DC to midPoint, please run this command on AD DC:

```
telnet {midpoint-host} {midpoint-port}
```

If blank page is shown, communication is established.

If you see error message, please modify your firewall rules or contact your network administrator.

Regarding the midPoint signing certificate authority trust relationship, open midPoint user interface in browser (like <https://midpoint.contoso.com/midpoin>). Regarding exact URI used for accessing midPoint user interface, contact your network administrator. When opening the user interface, there cannot be any warning in reference to signing certification authority, certificate validity etc.

2.1.3 User in midPoint

To create user in midPoint for midpointUpdatingService, log into your already installed midPoint where you need to forward password changes from Active Directory. At first, import role template over menu → Import object → Embedded editor this XML (application role for AD password agent user – “ad-agent-role”):

Click button “Import object”.

Then, create new user over menu → Users → New user, set at least these properties:

- Name = “ad-agent” (without “)
- Administrative status = Enabled
- Password – we recommended to use strong password compliant with midpoint security policy for service accounts “never expire”
- Tab Assignments → New assignments, find and select role “ad-agent-role” and click button “Add”
- Click button “Save”.

Created midpoint user with noted password will be used by “midpointUpdatingService” to connect midPoint IDM Rest service.

2.1.4 midpoint SSL TRUST configuration

We recommend midPoint being configured with HTTPS secure interface. If so, download certificate from your midPoint server (if is self-signed) or get his root CA certificate to allow trust relationship to be established. Format of the certificate can be in Base-64 encoded X.509 cert form (PEM format).

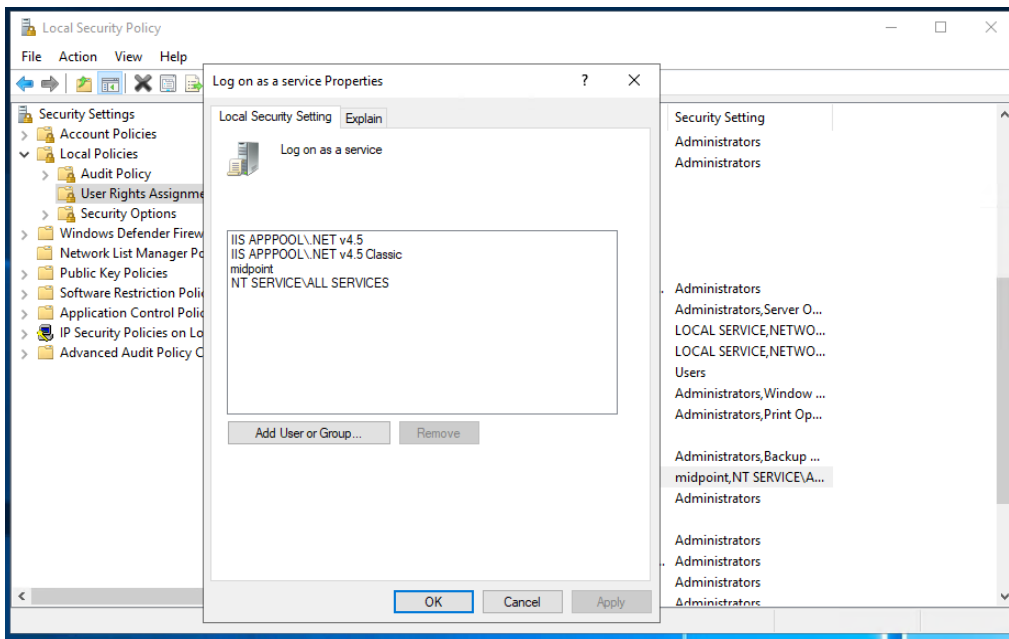
Install certificate to domain controller, where agent will run, in Local Machine store and place it into “[Trusted Root Certification authorities](#)” folder.

2.1.5 User in Active Directory

Create a user in Active directory domain for “midpointUpdatingService”. You can use “[Active Directory Users and Computers](#)” tool. Browse the directory tree to the container, where you prefer to create the service account, and click “[Create a new user in the current container](#)”. Set at least these properties:

- User logon name = “[midpoint](#)”
- Strong password based on your domain policy
- Uncheck “User must change password at next logon”
- Check “User cannot change password”
- Check “Password never expires”
- Uncheck “Account is disabled”
- Group “Domain User”
- Group “Domain administrators”
- Rights to log on as a service

Set-up user rights to log on as a service over “[Local Security Policy](#)”, “[Local Policies](#)”, “[User Rights Assignments](#)”, “[Log on as a service Properties](#)” for user “[midpoint](#)”:



After the installation succeeded, you can turn the service to use a system account.

2.1.6 Service client certificate generation

Service client certificate is used to secure midPoint server / domain controller communication.

To generate and import service client certificate run command:

New-SelfSignedCertificate -DnsName {dns-name-for-ad-dc} -CertStoreLocation "cert:\LocalMachine\My"

Your Certificate SubjectDN/Path to certificate (ServiceClientCertificate) will be "CN={dns-name-for-ad-dc}" – prefix "CN=" is required.

Optionally, you can use existing certificate from domain controller local machine certificate store. Using existing certificate prerequisite is to identify it properly for configuration option. The recommended approach is to use powershell command as administrator to list the subject of certificates in myComputer personal store in form expected by install procedure. The command is: **"Get-ChildItem -path cert:\LocalMachine\My"**

```
PS C:\Users\Administrator> Get-ChildItem -path cert:\LocalMachine\My

PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\My

Thumbprint                               Subject
-----
EC5E42A32B602552A497F0D1B25D4A192AE2B462 C=SK, L=Bratislava, OU=Prevádzka IT, O=LNHome, CN=dc1.lnhome.sk
```

Reference for setting "MidpointSSL" config option:

MidpointSSL Value	Description
0	No SSL – no client certificate present, midPoint is HTTP only (for testing or development purposes).
1	TLS 1.2 (Certificate is stored in LocalMachine Private store and identified by Distinguished Name in parameter "ServiceClientCertificate")

2.2 Installer installation

On the each Domain Controller in target domain open CMD window as an Administrator (elevated privileges mode as Domain Administrator) and run:

```
install.cmd
```



Accept the terms of License Agreement and click "Install"

The screenshot shows a 'Configuration' window with the following fields and values:

- MidPoint Base URL:
- Username:
- Password:
- MidPoint Queue Identifier:
- Number of attempts on MidPoint call:
- Logging level 0-verbose to 4-error only:
- Log storage path:
- SSL Setting 0-HTTP, 1-HTTPS/TLS1.2 (Certificate in I...:
- Service Client Certificate SubjectDN / Path to certific...:
- Midpoint updating service context account:
- Midpoint updating service context account password:

At the bottom, there are 'Cancel' and 'Next' buttons.

Set-up configuration options:

Parameter	Registry key	Description
MidPoint Base URL	MidpointBaseUrl	Midpoint Base URL (http://<your.midpoint.url>:port/midpoint)
MidPoint Account Username	MidpointAuthUser	Midpoint username (for example "midpoint-agent")
MidPoint Account Password	MidpointAuthPwd	Midpoint password
MidPoint Queue Identifier	QueueFolder	do not change the default setting if there is not more than one Midpoint synchronized from the same DC
Number of attempts on MidPoint call	RetryCount	max 500 for performance reasons

Logging level	MidpointServiceLogLevel	0-debug, 1-info, 2-warning, 3-Error, 4-Fatal error only
Log storage path	MidpointServiceLogPath	Path for Logs of midpoint Updating Service. If it does not exists, it has to be created manually, or los will be placed in default directory .\Logs
SSL Setting	MidpointSSL	0-HTTP only, 1-HTTPS/TLS 1.2 with certificate in Local Computer repository, 2- HTTPS/TLS 1.2 with certificate in file X.509
Certificate SubjectDN/Path to certificate	ServiceClientCertificate	mode 1 contains AD server certificate SubjectDN of the certificate in form CN=xxx.yyy.zzz in SSL (prefix "CN=..." is required!), see also 2.1.6 Service client certificate generation mode 2 contains Path and full name of the X.509 CER file.
Midpoint updating service context account	-	Domain user account used for running the midpointUpdatingService (for example "midpoint-agent")
Midpoint updating service context account password	-	Domain user used for running the midpointUpdatingService password

Click "Next" and wait for configuration to finish.

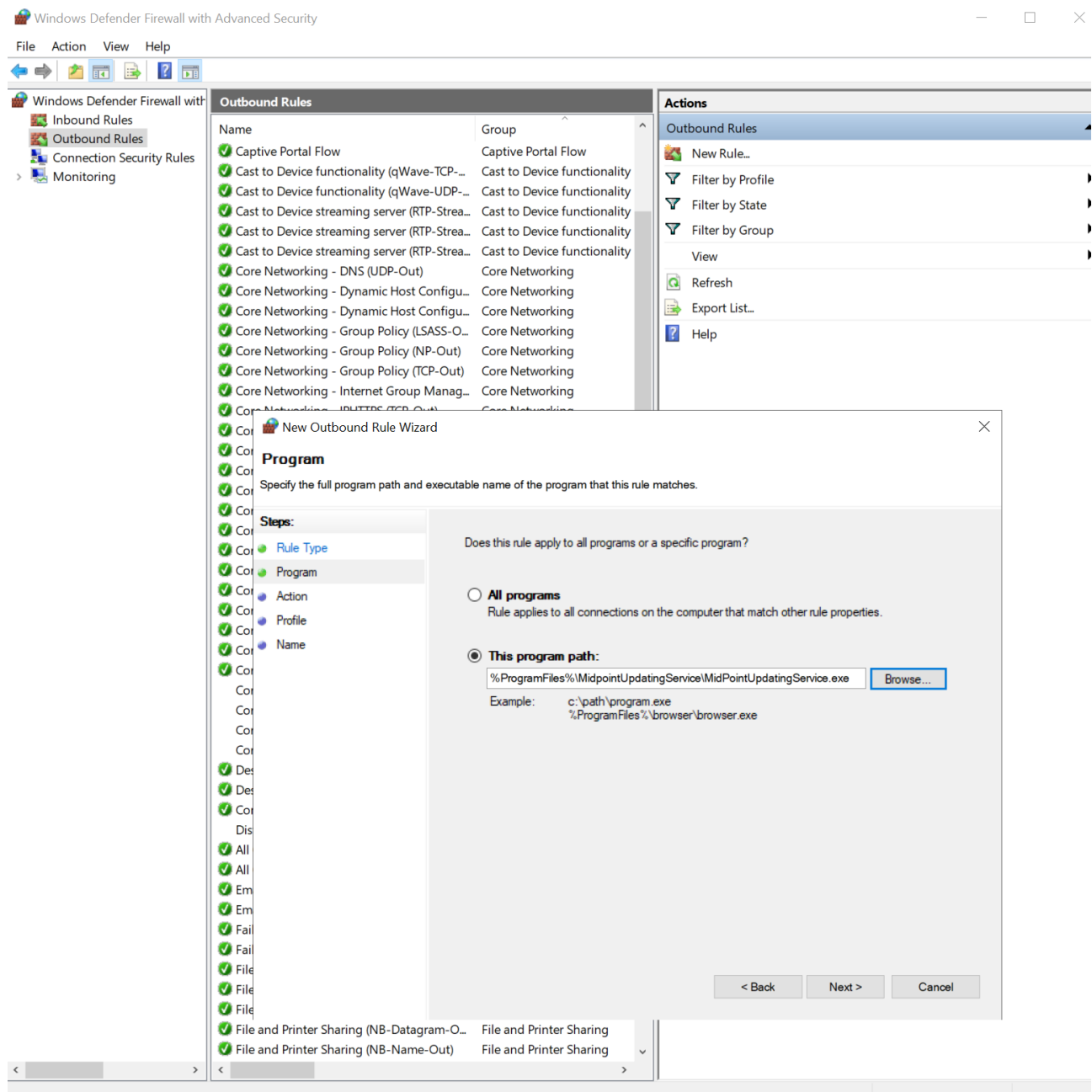
Restart Domain Controller to activate ADPasswordFilter.dll and verify, if the service named MidpointUpdatingService is running.

Manual check (after installation):

- Installed Binaries
 - C:\Program Files\midpointUpdatingService*.*
 - C:\Windows\System32\ADPasswordFilter.dll
- New windows registry keys:
 - HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages\ADPasswordFilter
 - HKLM\SYSTEM\CurrentControlSet\Services\MidpointUpdatingService
 - HKLM\SOFTWARE\ADPasswordFilter
- New windows service
 - MidpointUpdatingService

3. Recommended postinstall activities

3.1 DC controller firewall configuration modification



New Outbound Rule Wizard



Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

Steps:

- Rule Type
- Program
- Action
- Profile
- Name

What action should be taken when a connection matches the specified conditions?

☒ **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

☐ **Allow the connection if it is secure**
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.
[Customize...](#)

☐ **Block the connection**

< Back Next > Cancel

New Outbound Rule Wizard



Profile

Specify the profiles for which this rule applies.

Steps:

- Rule Type
- Program
- Action
- Profile
- Name

When does this rule apply?

☒ **Domain**
Applies when a computer is connected to its corporate domain.

☒ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

☒ **Public**
Applies when a computer is connected to a public network location.

< Back Next > Cancel

New Outbound Rule Wizard

Name

Specify the name and description of this rule.

Steps:

- Rule Type
- Program
- Action
- Profile
- Name**

Name:





Description (optional):

3.2 Elimination of Password change cycling

By design principle the password changed from MS AD is changed in midPoint IDM and immediately published to MS AD. Than in MS AD despite the fact of same value, within same minute as previous change, the password is accepted and pushed back to midPoint IDM bz password agent. IDM midPoint is not accepting the change due two facts (same value as previous value of password, too frequent change) what is resulting in failure. Solution is in limiting publishing the changed password from midPoint to MS AD for changes by any user except password agent user. Elimination of password change cycling below was tested with midpoint version 4.8. Under different circumstances it may be needed to apply different approach, please refer to [midpoint documentation](#). Example below is limited example for chosen password agent name "pwdagent". It does not have to fit your environment.

```
<credentials>
  <password>
    <outbound>
      <strength>strong</strength>
      <description>Do not process changed password by actor named pwdagent</description>
      <condition>
        <script>
          <code><![CDATA[
            return (actor?.name.orig != "pwdagent")
          ]]></code>
        </script>
      </condition>
    </outbound>
  </password>
</credentials>
```

After deploying elimination of password change cycling, the password change is successfully completed as in MS AD, as in midpoint IDM for any user except password agent user. The password cycling can be observed in history of user, like in this screen shot below

Time	Initiator	Event Type	Channel	Outcome	
2024-01-09T13:56:31.933Z	ad-pwd-agent	 Modify object	REST	Success	View object data View object xml
2024-01-09T13:56:31.774Z	ad-pwd-agent	 Modify object	REST	Success	View object data View object xml
2024-01-02T10:37:24.271Z	administrator	 Modify object	Reconciliation	Success	View object data View object xml
2024-01-02T10:37:23.836Z	administrator	 Modify object	Reconciliation	Success	View object data View object xml

3.3 Password agent authorization

It make sense to limit password agent user account for password change only by using midPoint authorization mechanismus. An example of password agent authorization role (It does not have to fit your environment).

```

<authorization>
  <name>rest-api-access</name>
  <decision>allow</decision>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-rest-3#all</action>
</authorization>

<authorization>
  <name>read-all-users</name>
  <decision>allow</decision>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#search</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#read</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#get</action>
  <object>
    <type>UserType</type>
  </object>
  <item>name</item>
  <item>description</item>
  <item>credentials</item>
</authorization>

<authorization>
  <name>modify-user-request</name>
  <decision>allow</decision>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#add</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#delete</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#modify</action>
  <phase>request</phase>
  <object>
    <type>UserType</type>
  </object>
  <item>name</item>
  <item>description</item>
  <item>credentials</item>
</authorization>

<authorization>
  <name>add-modify-shadow-request</name>
  <decision>allow</decision>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#add</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#read</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#get</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#search</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#modify</action>
  <phase>request</phase>
  <object>
    <type>ShadowType</type>
  </object>
</authorization>

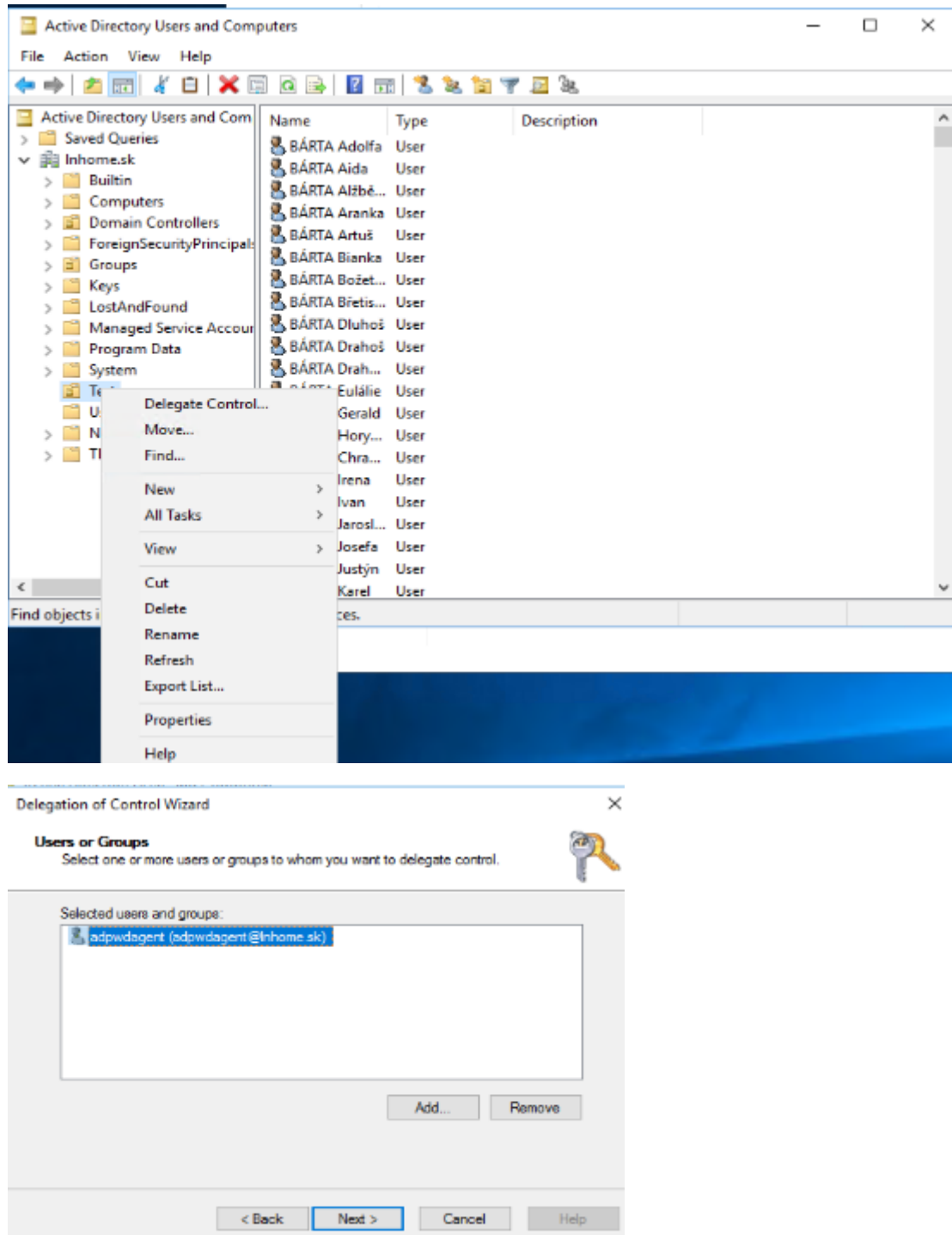
<authorization>
  <name>add-modify-execution</name>
  <decision>allow</decision>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#add</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#read</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#get</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#search</action>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#modify</action>
  <phase>execution</phase>
  <object>
    <type>ShadowType</type>
  </object>
  <object>
    <type>ResourceType</type>
  </object>
  <object>
    <type>UserType</type>
  </object>
</authorization>

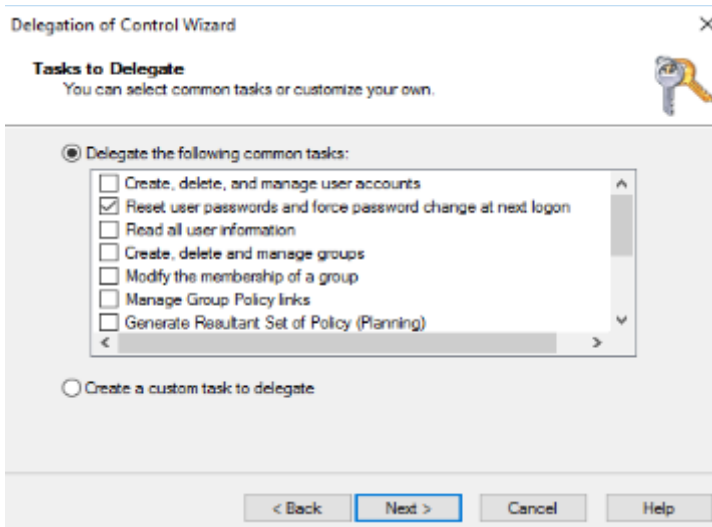
<authorization>
  <name>self-shadow-credentials-request</name>
  <decision>allow</decision>
  <description>
    Allow to modify credentials of all users accounts.
    Note that this is a request phase authorization. It also requires corresponding execution-phase authorization.
  </description>
  <action>http://midpoint.evolveum.com/xml/ns/public/security/authorization-model-3#changeCredentials</action>
  <phase>request</phase>
  <object>
    <type>ShadowType</type>
  </object>
  <item>credentials</item>
</authorization>

```

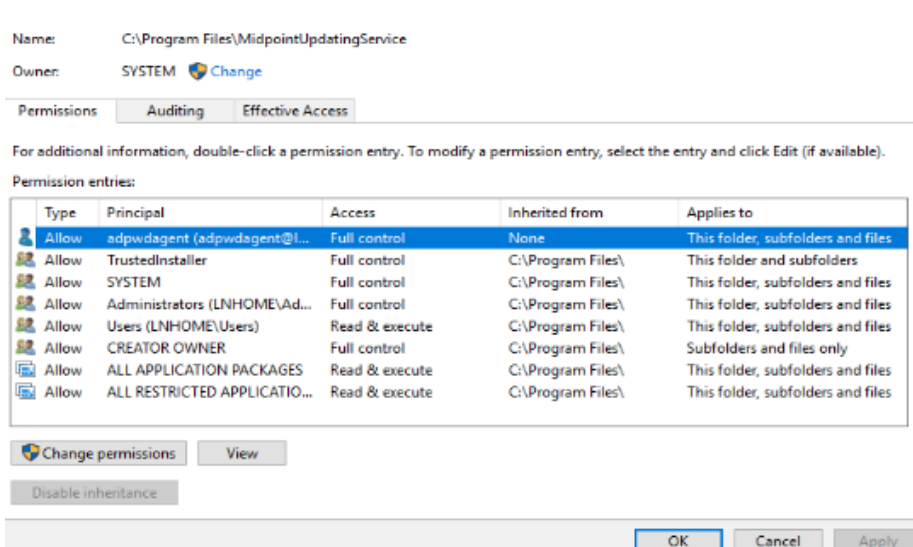
3.4 Limiting password agent MS AD user object rights

Password agent user object needs delegated administration over managed users to allow change of their password.





And we must assign permission to all AD Password Agent working directories



Name: C:\Program Files\ADPasswordAgent

Owner: SYSTEM [Change](#)

Permissions: Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Allow	adpwdagent (adpwdagent@i...	Full control	None	This folder, subfolders and files
Allow	TrustedInstaller	Full control	C:\Program Files\	This folder and subfolders
Allow	SYSTEM	Full control	C:\Program Files\	This folder, subfolders and files
Allow	Administrators (LNHOME\Ad...	Full control	C:\Program Files\	This folder, subfolders and files
Allow	Users (LNHOME\Users)	Read & execute	C:\Program Files\	This folder, subfolders and files
Allow	CREATOR OWNER	Full control	C:\Program Files\	Subfolders and files only
Allow	ALL APPLICATION PACKAGES	Read & execute	C:\Program Files\	This folder, subfolders and files
Allow	ALL RESTRICTED APPLICATIO...	Read & execute	C:\Program Files\	This folder, subfolders and files

[Change permissions](#) [View](#)

[Disable inheritance](#)

[OK](#) [Cancel](#) [Apply](#)

Name: C:\ProgramData\Midpoint.ADPassword.Queue.Heap

Owner: adpwdagent (adpwdagent@lnhome.sk) [Change](#)

Permissions: Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Allow	adpwdagent (adpwdagent@i...	Full control	None	This folder, subfolders and files
Allow	SYSTEM	Full control	C:\ProgramData\	This folder, subfolders and files
Allow	Administrators (LNHOME\Ad...	Full control	C:\ProgramData\	This folder, subfolders and files
Allow	adpwdagent (adpwdagent@i...	Full control	C:\ProgramData\	This folder only
Allow	CREATOR OWNER	Full control	C:\ProgramData\	Subfolders and files only
Allow	Users (LNHOME\Users)	Read & execute	C:\ProgramData\	This folder, subfolders and files
Allow	Users (LNHOME\Users)	Write	C:\ProgramData\	This folder and subfolders

[Change permissions](#) [View](#)

[Disable inheritance](#)

[OK](#) [Cancel](#) [Apply](#)

And it must have allowance to run as service

Local Group Policy Editor

File Action View Help

Local Computer Policy

- Computer Configuration
 - Software Settings
 - Windows Settings
 - Name Resolution Policy
 - Scripts (Startup/Shutdown)
 - Deployed Printers
 - Security Settings
 - Account Policies
 - Password Policy
 - Account Lockout Policy
 - Windows Firewall with Advanced Security
 - Network List Manager Policies
 - Software Key Policies
 - Software Restriction Policies
 - Application Control Policies
 - IP Security Policies on Local Computer
 - Advanced Audit Policy Configuration
 - Local Policies
 - User Rights Assignment
 - Security Options
 - Administrative Templates
 - User Configuration
 - Software Settings
 - Windows Settings
 - Administrative Templates

Policy

- Access Credential Manager as a trusted caller
- Access this computer from the network
- Act as part of the operating system
- Add workstations to domain
- Adjust memory quotas for a process
- Allow log on locally
- Allow log on through Remote Desktop Services
- Back up files and directories
- Bypass traverse checking
- Create a pagfile
- Create a token object
- Create global objects
- Create permanent shared objects
- Create symbolic links
- Debug programs
- Deny access to this computer from the network
- Deny log on as a batch job
- Deny log on as a service
- Deny log on locally
- Deny log on through Remote Desktop Services
- Enable computer and user accounts to be automatically authenticated
- Force shutdown from a remote system
- Generate security audits
- Change the system time
- Change the time zone
- Impersonate a client after authentication
- Increase a process working set
- Increase scheduling priority
- Load and unload device drivers
- Lock pages in memory
- Log on as a batch job
- Log on as a service
- Manage auditing and security log
- Modify an object label
- Modify firmware environment values

Security Setting

- Everyone,Authenticated Users
- Authenticated Users
- LOCAL SERVICE,NETWORK SERVICE
- Administrators,Account...
- Administrators
- Administrators,Server O...

Log on as a service Properties

Local Security Setting Explain

Log on as a service

LNHOME\adpwdagent

NT SERVICE\ALL SERVICES

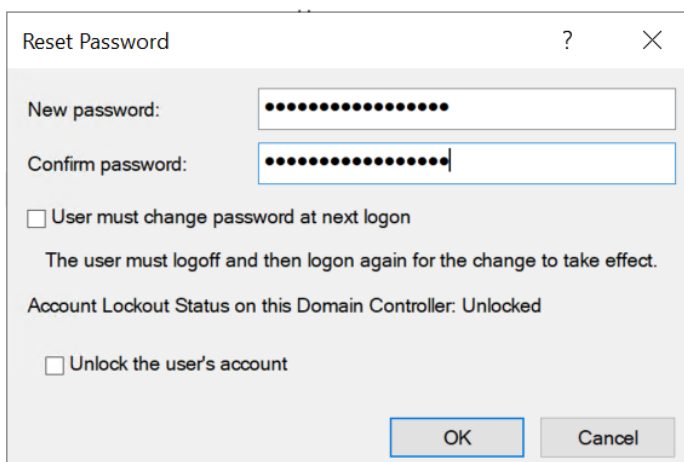
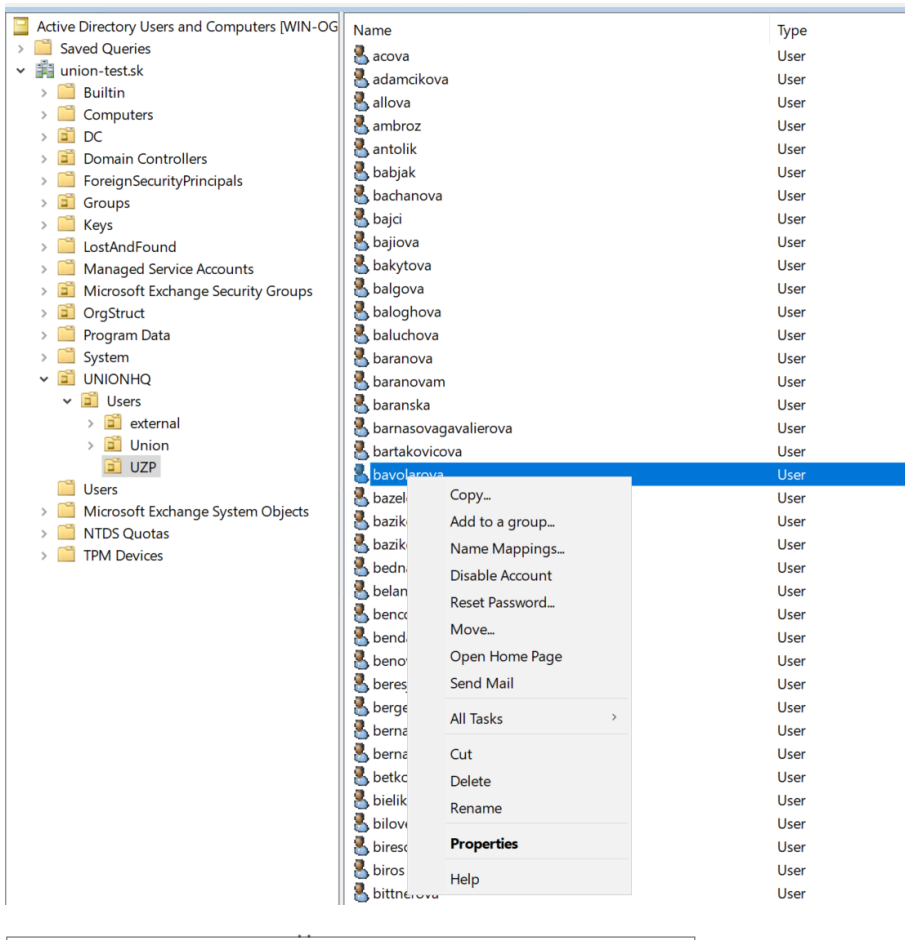
[Add User or Group...](#) [Remove](#)

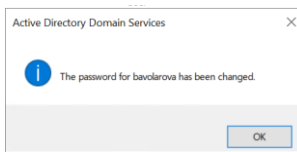
[OK](#) [Cancel](#) [Apply](#)

4. Testing / Troubleshooting

4.1 AD Basic test

Change password in AD for user that already exists in midPoint.





Check in midPoint user object history for user with changed password. In case of error, the reason is visible in history details.

Time	Initiator	Event Type	Channel	Outcome		
2024-01-09T09:49:49.409Z	ad-pwd-agent	Modify object	REST	Fatal Error	View object data	View object url
2024-01-09T09:49:49.051Z	ad-pwd-agent	Modify object	REST	Fatal Error	View object data	View object url
2024-01-09T09:49:48.697Z	ad-pwd-agent	Modify object	REST	Success	View object data	View object url
2024-01-09T09:49:48.636Z	ad-pwd-agent	Modify object	REST	Success	View object data	View object url

UNION DEV Inalogy: Audit Log Details	
Timestamp	2024-01-09T09:49:49.051Z
Event Identifier	1704793789051-50089-1
Event Type	MODIFY_OBJECT
Event Stage	EXECUTION
Initiator	ad-pwd-agent
Attorney	
Effective Principal	ad-pwd-agent
Effective Privileges Modification	
Target ref.	bavolarova
Target Owner ref.	
Result	
Outcome	FATAL_ERROR
Session Identifier	1704793788903-50089-1
Task Identifier	1704793788903-50089-1
Task oid	
Request Identifier	05034d54-14fe-4f2c-b025-e5c160686acf
Host Identifier	10.30.0.35
Node	DefaultNode
Remote Host	10.30.0.36
Channel	http://midpoint.evolveum.com/xml/ns/public/common/channels-3#rest
Parameter	
Message	Provided password does not satisfy the policies: Minimal age was not yet reached. The value was recently used.

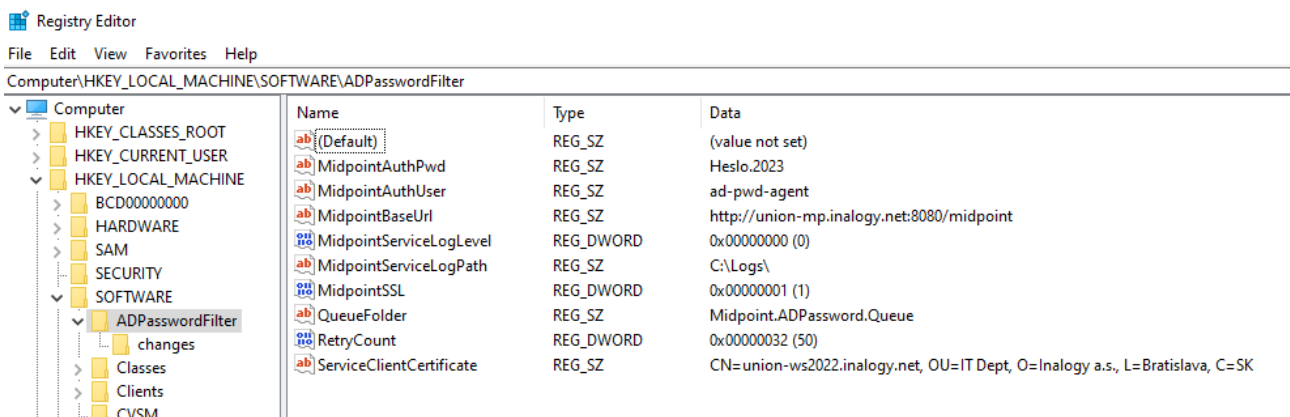
4.2 Troubleshooting installation with installer

Verify [Installer.log](#) in install directory for errors. In case for install procedure is used install.cmd command (part of insalation files), install log noctains all details entered as all issues encountered during install.

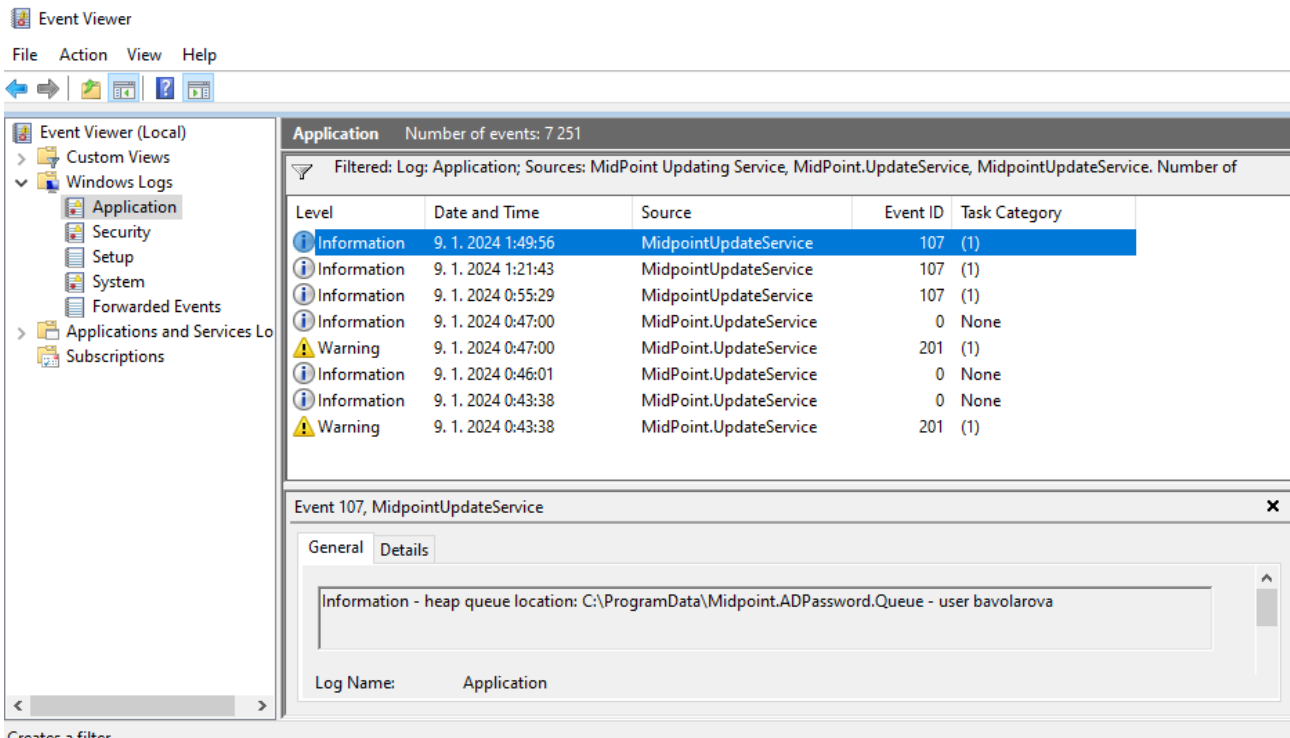
4.3 ADPasswordAgent troubleshooting

All events are recorded into Windows Application EventLog as Source="MidpointUpdateService"

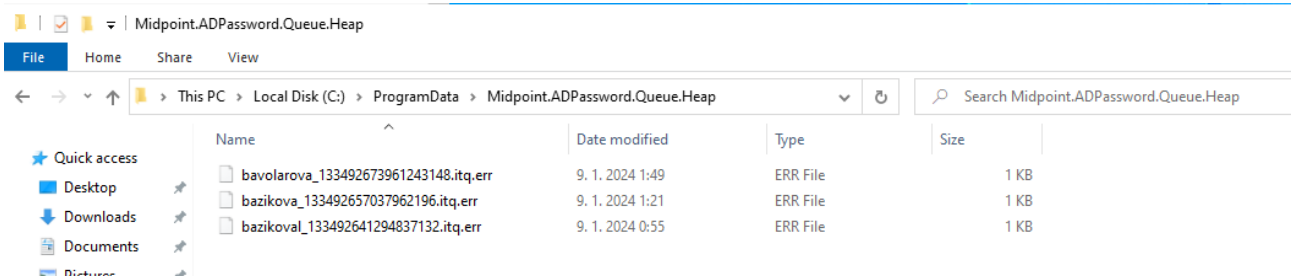
HKLM\SOFTWARE\ADPasswordFilter\MidpointServiceLogLevel



0-debug, 1-info, 2-warning, 3- Error to 4- Fatal error only



Not processed password changes are stored in folder

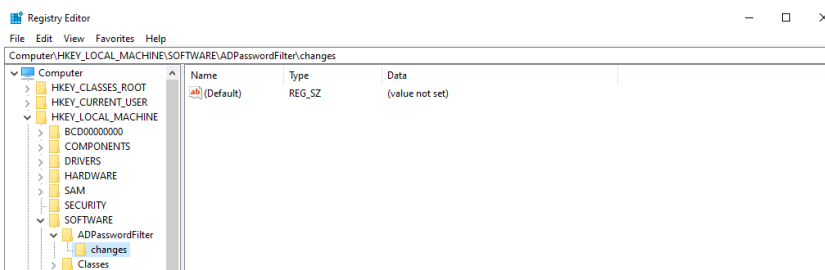


The content of not processed folder is reviewed periodically by password agent service. It makes sense to keep retry count quite low, as each retry doubles the delay between retry. Trying to process a password changed a month ago does not make sense. Recommended value for retry is about 10 (delayed processing by a day).

Common cause of not processed password change is different password policy in MS AD than in midPoint IDM. As example on page 14 is visible cause of changing password too frequently. That exact example has root cause in supporting changing password as often as client prefer to on MS AD site and on midPoint IDM site limiting the change not often than once a day.

4.4 Location of registry password hive

[HKEY_LOCAL_MACHINE\SOFTWARE\ADPasswordFilter\changes]



4.5 Location of file storage

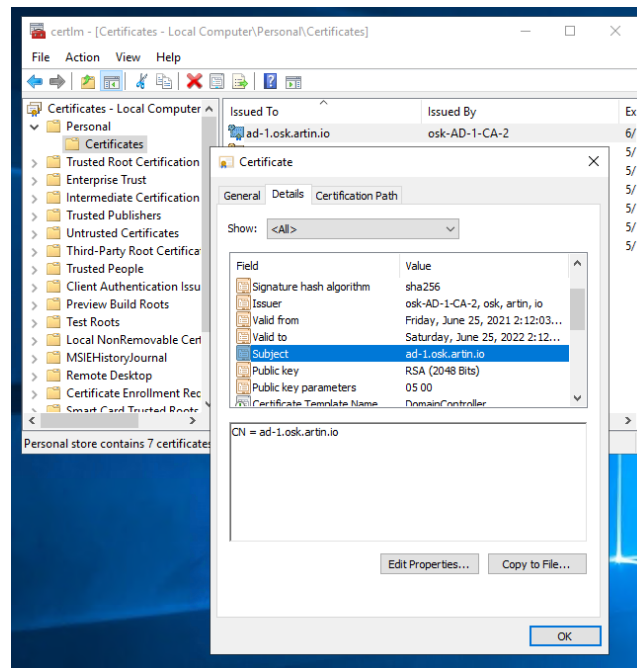
C:\ProgramData\Midpoint.ADPassord.Queue.Heap\...

One file for one user with timestamp when his password was changed and only under condition of communication failure between domain controller and midpoint.

File exists only as local persistence store for time while new password is not sent to midPoint.

4.6 Location of server certificate

Run `certmgr.msc`



4.7 midPointUpdatingService Logging troubleshooting

All events are recorded into Windows File system <midPointUpdatingService.home>\Logs\EventLog.txt















HKLM\SOFTWARE\ADPasswordAgent\MidpointServiceLogLevel

0-debug, 1-info, 2-warning, 3- Error to 4- Fatal error only

Name	Type	Data
(Default)	REG_SZ	(value not set)
Agent	REG_SZ	C:\Program Files\ADPasswordAgent\ADPasswordAgent.exe
AgentLogging	REG_DWORD	0x00000000 (0)
MidpointAuthPwd	REG_SZ	aiseeM4ahmei1aem
MidpointAuthUser	REG_SZ	administrator
MidpointBaseUrl	REG_SZ	https://midpoint-1.osk.artin.io:8443/midpoint
MidpointServiceLogLevel	REG_DWORD	0x00000000 (0)
MidpointServiceLogPath	REG_SZ	Logs\
MidpointSSL	REG_DWORD	0x00000001 (1)
MidpointUseOnlyHeap	REG_DWORD	0x00000000 (0)
QueueFolder	REG_SZ	Midpoint.ADPassword.Queue
QueueWaitSeconds	REG_DWORD	0x0000001e (30)
RetryCount	REG_DWORD	0x00000032 (50)
ServiceClientCertificate	REG_SZ	CN=ad-1.osk.artin.io

midPointUpdatingServiceLogging – changing log location

HKLM\SOFTWARE\ADPasswordAgent\MidpointServiceLogPath

Name	Type	Data
 (Default)	REG_SZ	(value not set)
 Agent	REG_SZ	C:\Program Files\ADPasswordAgent\ADPasswordAgent.exe
 AgentLogging	REG_DWORD	0x00000000 (0)
 MidpointAuthPwd	REG_SZ	aiseeM4ahmei1aem
 MidpointAuthUser	REG_SZ	administrator
 MidpointBaseUrl	REG_SZ	https://midpoint-1.osk.artin.io:8443/midpoint
 MidpointServiceLogLevel	REG_DWORD	0x00000000 (0)
 MidpointServiceLogPath	REG_SZ	Logs\
 MidpointSSL	REG_DWORD	0x00000001 (1)
 MidpointUseOnlyHeap	REG_DWORD	0x00000000 (0)
 QueueFolder	REG_SZ	Midpoint.ADPassword.Queue
 QueueWaitSeconds	REG_DWORD	0x0000001e (30)
 RetryCount	REG_DWORD	0x00000032 (50)
 ServiceClientCertificate	REG_SZ	CN=ad-1.osk.artin.io