



MyID PIV
Version 12.9

Mobile Authentication

Lutterworth Hall, St Mary's Road, Lutterworth, Leicestershire, LE17 4PS, UK
www.intercede.com | info@intercede.com | [@intercedemyid](https://twitter.com/intercedemyid) | +44 (0)1455 558111

Copyright

© 2001-2023 Intercede Limited. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished exclusively under a restricted license or non-disclosure agreement. Copies of software supplied by Intercede Limited may not be used resold or disclosed to third parties or used for any commercial purpose without written authorization from Intercede Limited and will perpetually remain the property of Intercede Limited. They may not be transferred to any computer without both a service contract for the use of the software on that computer being in existence and written authorization from Intercede Limited.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Intercede Limited.

Whilst Intercede Limited has made every effort in the preparation of this manual to ensure the accuracy of the information, the information contained in this manual is delivered without warranty, either express or implied. Intercede Limited will not be held liable for any damages caused, or alleged to be caused, either directly or indirectly by this manual.

Licenses and Trademarks

The Intercede[®] and MyID[®] word marks and the MyID[®] logo are registered trademarks of Intercede in the UK, US and other countries.

Microsoft and Windows are registered trademarks of Microsoft Corporation. Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such. All other trademarks acknowledged.

Apache log4net

Copyright 2004-2021 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<https://www.apache.org/>).

Conventions used in this document

- Lists:
 - Numbered lists are used to show the steps involved in completing a task when the order is important.
 - Bulleted lists are used when the order is unimportant or to show alternatives.
- **Bold** is used for menu items and for labels.
For example:
 - Record a valid email address in '**From**' email address.
 - Select **Save** from the **File** menu.
- *Italic* is used for emphasis:
For example:
 - Copy the file *before* starting the installation.
 - Do *not* remove the files before you have backed them up.
- ***Bold and italic*** hyperlinks are used to identify the titles of other documents.
For example: "See the ***Release Notes*** for further information."
Unless otherwise explicitly stated, all referenced documentation is available on the product installation media.
- A `fixed width` font is used where the identification of spaces is important, including filenames, example SQL queries and any entries made directly into configuration files or the database.
- **Notes** are used to provide further information, including any prerequisites or configuration additional to the standard specifications.
For example:
Note: This issue only occurs if updating from a previous version.
- **Warnings** are used to indicate where failure to follow a particular instruction may result in either loss of data or the need to manually configure elements of the system.
For example:
Warning: You must take a backup of your database before making any changes to it.

Contents

Mobile Authentication	1
Copyright	2
Conventions used in this document	3
Contents	4
1 Introduction	5
2 MyID Authenticator	6
2.1 Setting up a credential profile	6
2.2 Issuing a MyID Authenticator ID	8
2.2.1 Requesting a MyID Authenticator ID for yourself	8
2.2.2 Requesting a MyID Authenticator ID for another user	8
2.2.3 Collecting a MyID Authenticator ID	9
2.2.4 Requesting a MyID Authenticator ID as a derived credential	9
3 MyID Verification Service	10
3.1 Installing the verification service	11
3.2 Configuring the MyID Verification Service	11
3.2.1 Configuring the client certificate	12
3.2.2 Configuring Firebase Cloud Messaging	13
3.2.3 Configuring the database	14
3.2.4 Configuring the firewall for outgoing mobile notifications	15
3.2.5 Configuring the firewall for incoming mobile notifications	15
3.2.6 Configuring the firewall for incoming MyID Authenticator app data	15
3.2.7 Disabling 2-way TLS for the internal authentication service	16
4 AD FS Adapter Mobile	17
4.1 Overview	17
4.2 AD FS Adapter Mobile prerequisites	18
4.2.1 Mutual TLS	19
4.3 Installing the AD FS Adapter Mobile	20
4.3.1 Uninstalling the AD FS Adapter Mobile	24
4.3.2 Notification display text	25
4.4 Managing the AD FS Adapter Mobile	25
4.4.1 Configuration file	25
4.4.2 Managing themes	27
4.5 AD FS Adapter Mobile logging	27
4.5.1 Setting up AD FS Adapter Mobile logging	27
4.5.2 Viewing the Windows Event log	28

1 Introduction

MyID[®] provides allows you to provide authentication using an app on your mobile device instead of using passwords, providing secure two-factor authentication to your critical systems.

The system has the following components:

- MyID Authenticator – an app installed on your mobile device.
See section [2](#), *MyID Authenticator* for details.
- MyID Verification Service – a pair of REST-based web services that allow an external system to initiate a mobile authentication, and a mobile device to return authentication data.
See section [3](#), *MyID Verification Service* for details.
- AD FS Adapter Mobile – a component that provides authentication for AD FS (Active Directory Federation Services) using the MyID Authenticator app and the MyID Verification Service.
See section [4](#), *AD FS Adapter Mobile* for details.

2 MyID Authenticator

The MyID Authenticator is an app that you can install on your mobile device to provide secure two-factor authentication to your systems.

When you attempt to access your secure system, the system contacts MyID through the MyID Verification Service, which sends a push notification to your mobile device. You can then authenticate to the device using biometrics or PIN, and the MyID Authenticator sends a message back to MyID through the MyID Verification Service confirming that you have authenticated your identity, at which point MyID informs the secure system, which provides you access.

The process for issuing an identity using the MyID Authenticator app is as follows:

1. Set up a credential profile for the mobile identity.

See section [2.1, *Setting up a credential profile*](#) for details.

2. Install the MyID Authenticator app on your mobile device.

For the current release, the MyID Authenticator is available for the following:

- iOS

The app is available on the App Store.

- Android

The app is available on the Google Play store.

3. Request a mobile identity using MyID Desktop or the Self-Service Request Portal.

See section [2.2.1, *Requesting a MyID Authenticator ID for yourself*](#), section [2.2.2, *Requesting a MyID Authenticator ID for another user*](#), and section [2.2.4, *Requesting a MyID Authenticator ID as a derived credential*](#) for details.

4. Collect the mobile identity using the MyID Authenticator app.

See section [2.2.3, *Collecting a MyID Authenticator ID*](#) for details.

Once you have issued an identity to your device, you can use it to authenticate to your secure system.

Note: You cannot install the MyID Authenticator and the Identity Agent apps on the same device at the same time; this is due to URL provisioning restrictions.

2.1 Setting up a credential profile

You must set up a credential profile for the MyID Authenticator ID.

To set up a credential profile:

1. In MyID Desktop, log on as an administrator.
2. From the **Configuration** category, click **Credential Profiles**.
3. Click **New**.
4. Type a **Name** for the credential profile.
For example, `Mobile Authentication ID`.
5. In **Card Encoding**, select **Identity Agent**.

If you want to request and collect your MyID Authenticator identity using the Self-Service Request Portal (SSRP), you must also select the **Derived Credential** option. For more information on using the SSRP, see the *Setting up the credential profiles for derived credentials* section in the [Derived Credentials Self-Service Request Portal](#) document.

6. In **Device Profiles**, from the **Card Format** drop-down list select **Mobile**.
7. Click **Next**.
8. Select one certificate.

The certificate must be capable of being used as a digital signature; for example, an authentication or signing certificate. This certificate is used to sign the communications between the MyID Verification Service and the MyID Authenticator App.

Note: There is an option to select a container for the certificate – do not select a container. For the MyID Authenticator app to use the certificate, it must be in the Intercede keystore, not the System Store, so leave the option as the default **Select a Container** option.
9. Click **Next**.
10. Select the roles associated with the credential profile.

See the *Linking credential profiles to roles* section in the [Administration Guide](#) for details.
11. Click **Next**.
12. Do not select any card layouts, then click **Next**.
13. Add any **Comments**, then click **Next** to complete the workflow.

2.2 Issuing a MyID Authenticator ID

You can request a MyID Authenticator ID for your own account within MyID Desktop, then scan a QR code using the MyID Authenticator app to collect the ID onto your mobile device. Alternatively, an operator can request a MyID Authenticator ID using MyID Desktop, which then sends an SMS or email message to the end user, who can click on a link in the message to open the MyID Authenticator app and collect the ID onto their mobile device.

2.2.1 Requesting a MyID Authenticator ID for yourself

To request a MyID Authenticator ID:

1. Log on to MyID Desktop.
2. From the **Mobile Devices** category, select **Request My ID**.
3. From the list of mobile credential profiles, select the credential profile that was set up for the MyID Authenticator app.

Note: The list of credential profiles also includes credential profiles that were set up for the Identity Agent app. Make sure you select the correct profile.

4. Click **Continue**.

Depending on how your system is set up, the notification is sent by email or SMS. Take a note of the one time password.

For more information, see the *Requesting a mobile ID for your own mobile device* section in the [Mobile Identity Management](#) guide.

Alternatively, click **QR Code** to display a QR code that you can scan using the MyID Authenticator app to begin the collection process.

5. The screen displays a QR code.

2.2.2 Requesting a MyID Authenticator ID for another user

To request a MyID Authenticator ID:

1. Log on to MyID Desktop.
2. From the **Mobile Devices** category, select **Request ID**.
3. Search for the person to whom you want to issue the ID, then select the person from the list of search results.
4. From the list of mobile credential profiles, select the credential profile that was set up for the MyID Authenticator app.

Note: The list of credential profiles also includes credential profiles that were set up for the Identity Agent app. Make sure you select the correct profile.

5. Click **Continue**.

6. The details of the notification are displayed on screen.

Depending on how your system is set up, the notification is sent by email or SMS, and the one time password is displayed on screen, sent in a separate SMS message, or not displayed until the request has been approved; see the *Requesting a mobile ID for another user* section in the [Mobile Identity Management](#) guide for details.

7. Click **Send**.

2.2.3 Collecting a MyID Authenticator ID

To collect a MyID Authenticator ID, you can:

- Open the MyID Authenticator app and scan the QR code that is displayed on screen in MyID Desktop.
- Click the link in the SMS message sent to your mobile device.
- Click the link in the email message sent to your mobile device.

Follow the instructions in the app to complete the collection of the MyID Authenticator ID.

2.2.4 Requesting a MyID Authenticator ID as a derived credential

If you have an existing PIV smart card, issued either by the current MyID system or by another system, you can use it to request and collect a MyID Authenticator ID as a derived credential using the Self-Service Request Portal.

You must edit the credential profile to include the **Derived Credential** option in the **Card Encoding** section. See the [Derived Credentials Self-Service Request Portal](#) for details.

Note: Currently, you cannot use the Self-Service Kiosk to request a MyID Authenticator ID as a derived credential.

3 MyID Verification Service

The MyID Verification Service comprises a pair of REST-based web services that allow an external system to initiate a mobile authentication and a mobile device to return authentication data.

You install the MyID Verification Service on the MyID web server, and it serves as a link between external systems, mobile apps, and the MyID system. The service uses the MyID authentication database for recording authentication attempts and storing messaging information.

The general process is as follows:

1. A user attempts to access an external system.
2. The external system requests authentication for the user by sending a message to the MyID Verification Service.
3. The MyID Verification Service sends a push notification to the user's registered MyID Authenticator app.
4. The user responds to the notification, and authenticates to their device using the mobile device's PIN or biometrics.
5. The MyID Authenticator app responds to the MyID Verification Service.
6. The MyID Verification Service responds to the external system, confirming that the user has authenticated using the mobile app.
7. The external system allows access to the user.

3.1 Installing the verification service

You must install the MyID Verification Service on the MyID web server. This ensures that it can communicate with the MyID server components and database.

Important: Before you run the installation program, make sure that the MyID Web Service user has been granted local log on privileges (**Allow log on locally** option in the **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment** section of the Group Policy).

Note: The MyID Verification Service requires the ASP.NET Core Hosting Bundle. See the *Hardware and software requirements* section in the [Installation and Configuration Guide](#) for details.

To install the MyID Verification Service, select the **MyID Verification Service** option from the Server Roles and Features screen of the MyID Installation Assistant.

By default, the MyID Verification Service is installed to the following location:

```
C:\Program Files\Intercede\
```

The installation program creates a folder called `MyIDMobileAuthenticator` within this location to store the web service files.

For more information about running the MyID Installation Assistant, see the *MyID Installation Assistant* section in the [Installation and Configuration Guide](#).

3.2 Configuring the MyID Verification Service

The MyID Verification Service comprises the following web services:

- `MobileAuthInternal`

This internal web service is used to initiate the authentication process; it must be called only by trusted systems. For this reason, the web service is installed in such a way that it requires 2-way TLS authentication. Any client that needs to call the internal web service must use a client certificate that is trusted by the IIS server on which the internal web service is running.

- `MobileAuthExternal`

This external web service is used for mobile devices to send back authentication data. As such, this service does not require the same level of security configuration.

3.2.1 Configuring the client certificate

This is applicable to the `MobileAuthInternal` service only.

In addition to setting up 2-way TLS, you must allow the allowed client certificates, as IIS does not allow the required fine-grained control. You must add the thumbprint of the client certificate to the `appsettings.Production.json` file in the web service folder.

This file is the override configuration file for the `appsettings.json` files for the service. If the file does not already exist, you must create it in the same folder as the `appsettings.json` file.

By default, this is:

```
C:\Program Files\Intercede\MyIDMobileAuthenticator\InternalWS\
```

If you are creating a new `appsettings.Production.json` file, include the following content:

```
{
  "MyID": {
    "AllowedClientCertThumbprints": [
      "Add accepted client cert hex thumbprint value(s) here"
    ]
  }
}
```

Replace the text with a list of certificate thumbprints that have been issued for client systems to use; for example:

```
"AllowedClientCertThumbprints": [
  "8dcd7b4f081143e1b9b0108bc88720ccd01fe163"
]
```

If you already have an existing `appsettings.Production.json` file, add the `AllowedClientCertThumbprints` entry to the `MyID` section.

You can find the certificate thumbprint in the Properties dialog of the certificate, on the **Details** tab, in the **Thumbprint** field.

Note: If you want to use an alternative authentication method in IIS instead of 2-way TLS, you must disable TLS in the configuration file. See section [3.2.7, Disabling 2-way TLS for the internal authentication service](#).

3.2.2 Configuring Firebase Cloud Messaging

This is applicable to the `MobileAuthInternal` service only.

The `MobileAuthInternal` web service uses Firebase Cloud Messaging (FCM) to send notification messages to MyID-provisioned mobile devices. To access FCM, the web service requires an authentication token; contact Intercede customer support to obtain your token file, quoting reference SUP-326.

Once you have received your Firebase token file:

1. Copy the file to the web service folder.

By default, this is:

```
C:\Program Files\Intercede\MyIDMobileAuthenticator\InternalWS\
```

2. Open a PowerShell window using the MyID web services user.

This must be the same user as the one used for the `MobileAuthInternalPool` IIS application pool used by the `MobileAuthInternal` web service.

3. Run the following PowerShell command:

```
.\DPAPIEncryptFile.ps1 firebase.oath.json
```

This creates a file called `firebase.oath.json.enc` and removes the original file.

Note: If Firebase token file you have been provided has a different name, specify that name instead of `firebase.oath.json`. The PowerShell script creates an encrypted version of the file with `.enc` appended to its filename; for example, if your token file is `mytoken.oath.json`, the script creates an encrypted file called `mytoken.oath.json.enc`. You must then update the `appsettings.Production.json` file in the `InternalWS` folder to specify this filename; for example:

```
{
  "MyID": {
    "FirebaseCredentialPath": "mytoken.oath.json.enc",
    "AllowedClientCertThumbprints": [
      "8dcd7b4f081143e1b9b0108bc88720ccd01fe163"
    ]
  }
}
```

3.2.3 Configuring the database

Important: The installation procedure currently sets up the password for SQL Authentication incorrectly; if you are using SQL Authentication, you *must* follow the instructions in section [3.2.3.1, Encrypted database passwords](#) below to log on as the MyID Authentication user and encrypt and store your database password for both the `InternalWS` and `ExternalWS` web services.

This is applicable to both the `MobileAuthInternal` service and the `MobileAuthExternal` web service.

The internal and external facing web services both require access to the MyID database and the specific authentication database.

You configure the databases when you install the web service; see section [3.1, Installing the verification service](#) for details.

The database settings are stored in the `appsettings.json` file in each web service folder.

Important: The connection settings are updated when you run the installation program. If you have made any manual changes to the `appsettings.json` file, these are overwritten by the values you provide in the installer.

By default, these folders are:

```
C:\Program Files\Intercede\MyIDMobileAuthenticator\InternalWS\
```

and:

```
C:\Program Files\Intercede\MyIDMobileAuthenticator\ExternalWS\
```

A section at the bottom of the configuration file contains connection details; for example, for a database using Windows authentication:

```
"ConnectionStrings": {  
  "MyIDDatabase": "Database=MyID; Server=localhost; Trusted_Connection = true;",  
  "MobileAuthenticatorDatabase": "Database=MobileAuthenticator; Server=localhost; Trusted_  
Connection = true;"  
},
```

By default, the connection strings are configured for a database that runs locally and uses a trusted connection. These entries must be configured to reference the server where each database is running, and the correct authentication parameters.

3.2.3.1 Encrypted database passwords

If you are using a user ID and password instead of a trusted connection, and you need to update the connection settings after installation, you must specify a password in the `appsettings.json` file.

You can use the Password Change Tool to update the password; see the *Working with SQL accounts* section in the [Password Change Tool](#) guide.

Alternatively, you can update the file manually; to do this, you must encrypt this password, and use the `PasswordDPAPI` parameter instead of the usual clear text `Password` parameter.

Log on to the server as the MyID Authentication user and use the supplied `DPAPIEncrypt.ps1` PowerShell script to encrypt your password; this takes a single parameter, the password, and returns a Base64-encoded encrypted password. For example:

```
PS C:\Projects> .\DPAPIEncrypt.ps1 p455w0rd
AQAAANCMnd8BFdER[... ]bq/L/gCw==
```

You can then use the Base64-encoded encrypted password in the `PasswordDPAPI` parameter; for example:

```
PasswordDPAPI=AQAAANCMnd8BFdER[... ]bq/L/gCw==
```

3.2.4 Configuring the firewall for outgoing mobile notifications

The `MobileAuthInternal` web service must be able to communicate with Google's Firebase Cloud Messaging service.

See the Google documentation for details of configuring the firewall:

firebase.google.com/docs/cloud-messaging/concept-options#messaging-ports-and-your-firewall

3.2.5 Configuring the firewall for incoming mobile notifications

Mobile devices using the MyID Authenticator app are required to receive push notifications are part of their normal process.

See the Apple website for details of configuring firewalls for devices to receive these notifications:

support.apple.com/en-gb/HT203609

3.2.6 Configuring the firewall for incoming MyID Authenticator app data

The `MobileAuthExternal` web service receives communications from the MyID Authenticator app running on mobile devices. This is used periodically to receive push notification tokens from the app, and every time the app is used to perform an authentication.

Any firewall configuration affecting the `MobileAuthExternal` web service must allow standard https traffic through to the port that the `MobileAuthExternal` web service has been configured to listen on; typically, this is port 443.

3.2.7 Disabling 2-way TLS for the internal authentication service

This is applicable to the `MobileAuthInternal` service only.

By default, the service is configured to require 2-way TLS, and to specify a client certificate thumbprint – see section 3.2.1, *Configuring the client certificate*.

If you want to configure your system to use an alternative authentication system in IIS, you must disable TLS in the `MobileAuthInternal` service configuration file.

To disable TLS, you must add the `EnableClientCertAuthentication` setting with a value of `false` to the `appsettings.Production.json` file in the web service folder; by default, this is:

```
C:\Program Files\Intercede\MyIDMobileAuthenticator\InternalWS\
```

Add the setting to the `MyID` section.

For example:

```
{
  "MyID": {
    "EnableClientCertAuthentication": "false",
  }
}
```

If this setting is missing, or set to any value other than `false`, then 2-way TLS is required.

4 AD FS Adapter Mobile

The AD FS Adapter Mobile provides Active Directory Federation Services (AD FS) with a mobile authentication mechanism, using a credential that has been provisioned to the mobile device from MyID, to authenticate to a Relying Party Trust. You can configure the AD FS Adapter Mobile as a primary or additional authentication step using the AD FS Manager Tool. You must deploy the AD FS Adapter Mobile to your AD FS servers. You can install the AD FS Adapter Mobile on a system running Windows Server 2019 or 2022.

Intercede also provides an AD FS Adapter for OAuth, which you can use for FIDO devices; see the *MyID AD FS Adapter OAuth* section in the [MyID Authentication Guide](#).

4.1 Overview

With the AD FS Adapter Mobile installed and configured on AD FS, providing either primary or additional authentication for a Relying Party Trust, a user starts the authentication process by trying to access the Relying Party Trust service when they enter their email address at the login screen.

AD FS asks the AD FS Adapter Mobile if the email address (or UPN) provided is one it recognizes. The AD FS Adapter Mobile passes the email address (or UPN) to the MyID Verification Service to see if it is recognized. The AD FS Adapter Mobile returns the result to AD FS. If the email address (or UPN) is recognized, AD FS starts the authentication process calling into the AD FS Adapter Mobile passing in the claim containing the user's email address (or UPN).

The AD FS Adapter Mobile then requests an identity for push notification from the MyID Verification Service for the user with that email address (or UPN). The MyID Verification Service responds by requesting a push notification be sent to the user's mobile device, returning an identity to the AD FS Adapter Mobile. The AD FS Adapter Mobile uses this identity to poll the MyID Verification Service to see if the user has performed the biometric authentication on their device and what the result was.

A push notification appears on the user's mobile device, notifying them that they need to perform an authentication to access the Relying Party Trust service.

When the user taps on the notification on their mobile, it opens the MyID Authenticator app, which prompts them to accept or reject the request.

It also shows a random four-digit verification code. The verification code is displayed on both the MyID Authenticator app and the AD FS login screen. The user can check that the codes match.

If they accept the request on the MyID Authenticator app, they are prompted to enter a PIN or perform a biometric operation to verify their identity. If the verification succeeds, the MyID Authenticator app signs the challenge contained in the notification from the MyID Verification Service, using the private key of the authentication certificate provisioned to the device from MyID, and returns the result to the MyID Verification Service.

The MyID Verification Service verifies the signed challenge response from the MyID Authenticator app and sets a flag indicating the outcome.

The AD FS Adapter Mobile polling of this flag ends with success, failure, rejected, or timeout, depending on the outcome on the MyID Authenticator app.

The polling result determines if the AD FS Adapter Mobile allows AD FS to perform the login.

4.2 AD FS Adapter Mobile prerequisites

The following prerequisites must be in place before you install the AD FS Adapter Mobile:

- Relying Party Trust

A Relying Party Trust must exist under **AD FS Management > AD FS > Relying Party Trusts** to add the AD FS Adapter as a primary or additional authentication method.

- Access Control Policy

A suitable access control policy for controlling access to the Relying Party Trust under **AD FS Management > AD FS > Access Control Policies**; for example, “Permit everyone and require MFA” if the AD FS Adapter is used as an additional authentication method, or “Permit everyone” if the AD FS Adapter is used as a primary authentication method.

- Authentication Certificate

An authentication certificate must exist to secure communication to the MyID Verification Service. This certificate with private key must be installed on the AD FS server. The installed location and thumbprint of this certificate is required when you run the installation program.

See section [4.2.1, Mutual TLS](#) for more information.

- AD FS Service Account

The AD FS service account must be a member of the “domain users”. The AD FS service account needs “log on as a service” permission. To set this option, from **AD FS Server Manager > Tools > Local Security Policy > Security Settings > Local Policies > User Rights Assignment > Log on as a service > Local Security Setting tab > Add User or Group**, add the AD FS service account user.

This is required to allow the AD FS Adapter to use two-way TLS to the MyID Verification Service.

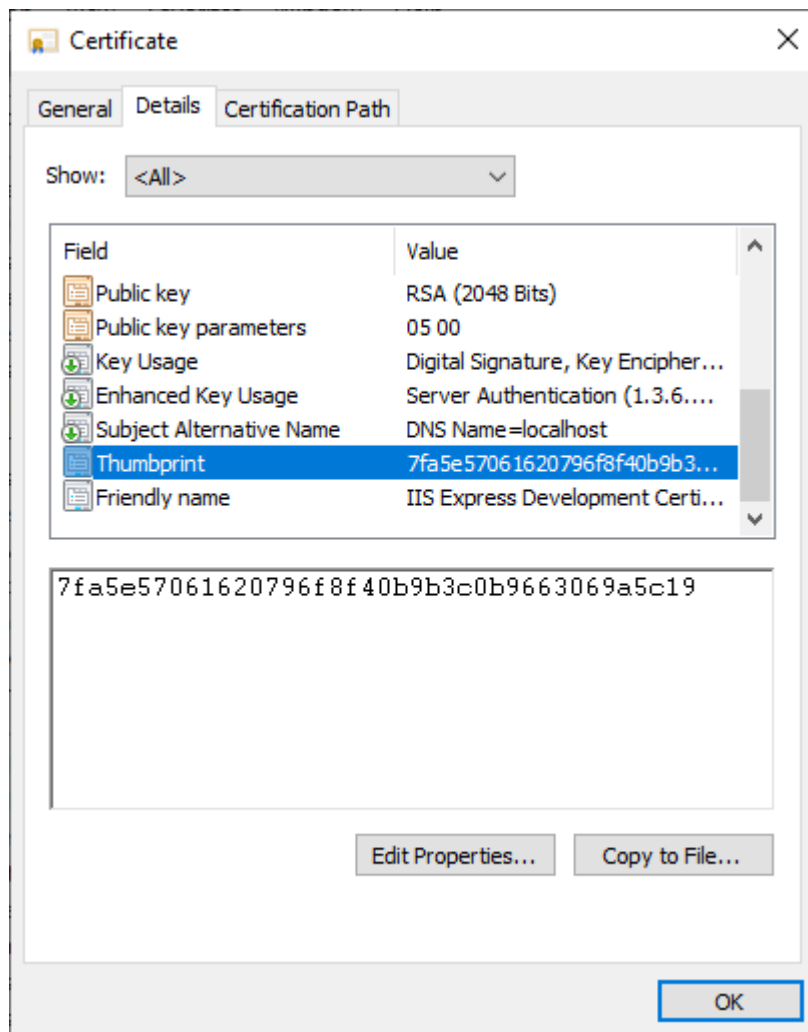
4.2.1 Mutual TLS

Two-way TLS is used to secure communication between the AD FS Adapter Mobile and the MyID Verification Service; this requires a client authentication certificate.

You must configure the MyID Verification Service `MobileAuthInternal` web service to allow this certificate; see section 3.2.1, [Configuring the client certificate](#) for details.

You must install the certificate with private key on AD FS before you run the AD FS Adapter Mobile installation program, which requires the store location, store name, and certificate thumbprint to identify the certificate.

You can find the certificate thumbprint in the Properties dialog of the certificate, on the **Details** tab, in the **Thumbprint** field.



4.3 Installing the AD FS Adapter Mobile

Before you install the AD FS Adapter Mobile, install the MyID Verification Service on the MyID web server; see section 3.1, *Installing the verification service*.

Make sure you have carried out the prerequisites for the AD FS Adapter before you run the installation program; see section 4.2, *AD FS Adapter Mobile prerequisites* for details.

You must install the AD FS Adapter Mobile on the AD FS server.

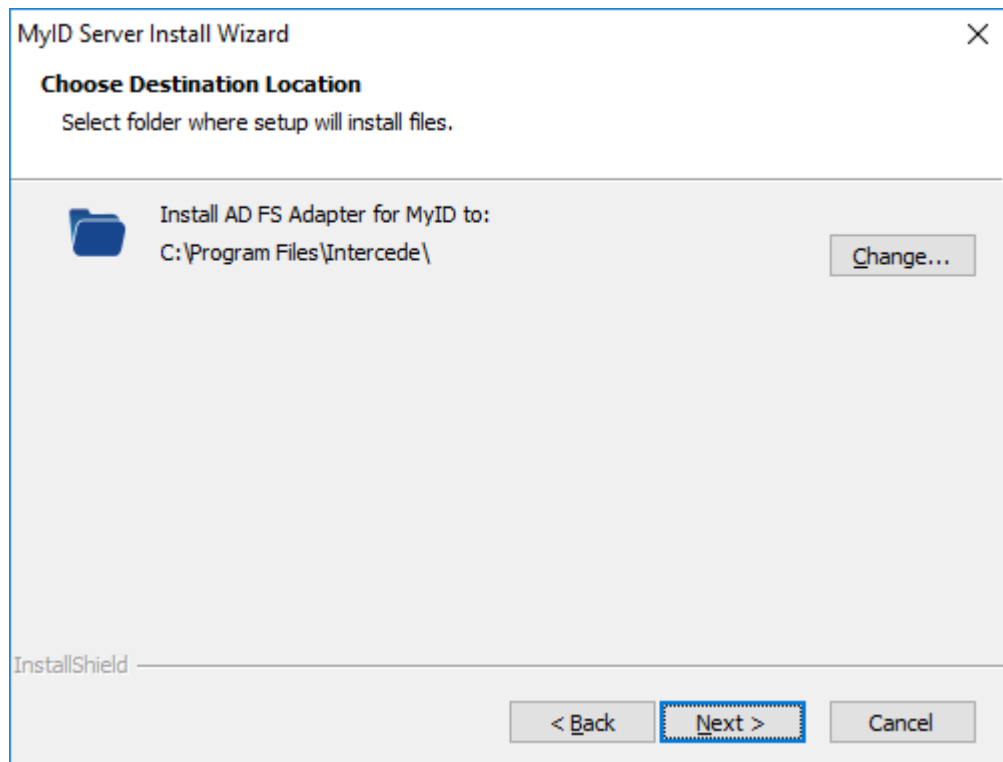
To install the adapter:

1. Copy the installation program onto the AD FS server.

The AD FS Adapter installation program is provided with the MyID installation media in the following folder:

```
\Authentication\AD FS Adapter for MyID\
```

2. Run the .msi installation program.
3. Click **Next** to begin.



4. Select the location for the AD FS Adapter.

By default, the AD FS Adapter is installed to the following location:

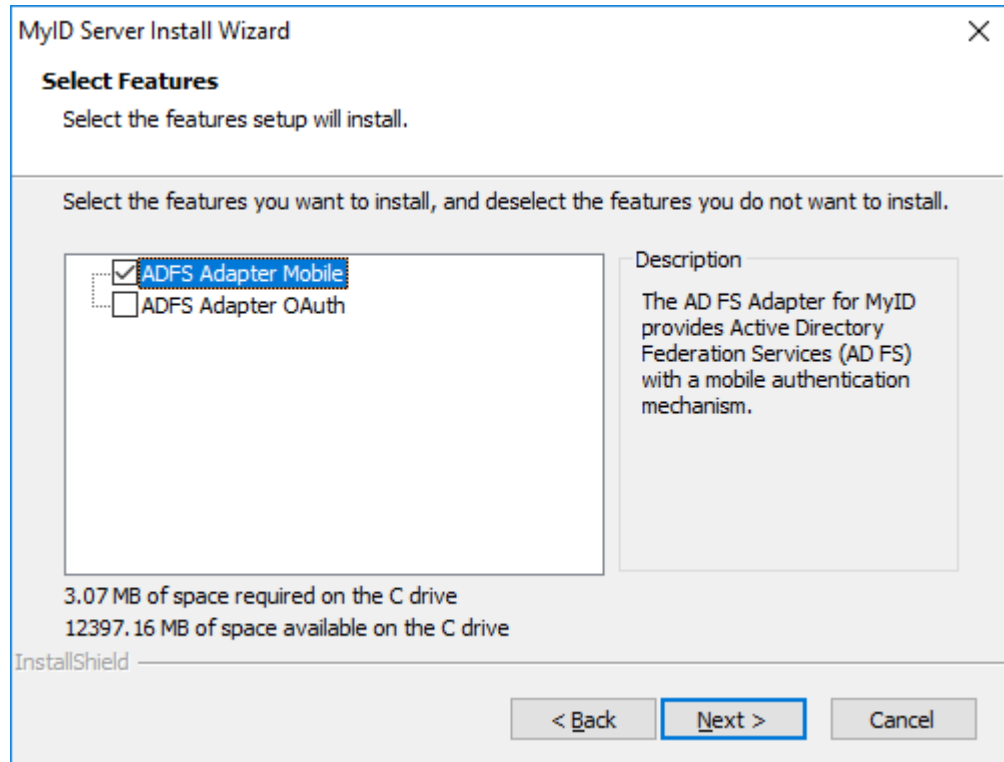
```
C:\Program Files\Intercede\
```

The installation program creates the following folders in this location:

- ADFS_Adapter_Mobile – contains the AD FS Adapter configuration files.
- Themes – contains the themes for the AD FS Adapter.

Note: The themes folder is shared with the AD FS Adapter OAuth, if you have it installed.

Click **Next**, and the Select Features screen appears.



5. Select the **ADFS Adapter Mobile** option.

For details of using the ADFS Adapter OAuth for FIDO devices, see the *Installing the ADFS Adapter OAuth* section in the [MyID Authentication Guide](#).

Click **Next**, and the Mobile ADFS Adapter Details screen appears.

6. Provide details of the MyID Verification Service:

- **URL of the MyID Verification Service** – the URL of the MyID Verification Service internal web service. The default is:

```
https://<MyID Web Service domain>/MobileAuthInternal/api/v1
```

Replace `<MyID Web Service domain>` with the name of your own MyID web server; for example:

```
https://myserver.example.com/MobileAuthInternal/api/v1
```

- **Push notification title as presented on the mobile notification screen** – the title of the notification that appears to the user on the mobile device when authentication is required.
- **Push notification description as presented on the mobile notification screen** – the text displayed in the notification that appears to the user on the mobile device when authentication is required.

See section [4.3.2, Notification display text](#) for information on how the push notification text is used on the MyID Authenticator app.

Click **Next**, and the Mutual TLS Certificate Details screen appears.

MyID Server Install Wizard

Mutual TLS Certificate Details

Please provide details of the certificate used to secure communication to the MyID Verification Service

Store Location:

Store Name:

mTLS Thumbprint:

InstallShield

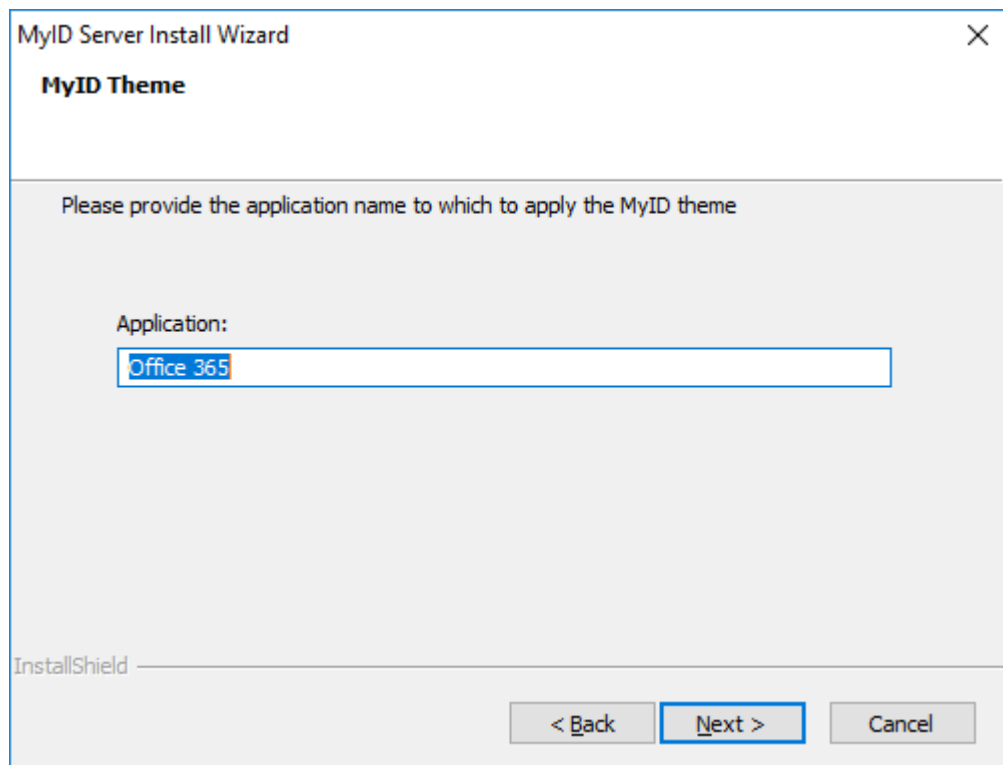
< Back Next > Cancel

7. Provide details of the mutual TLS certificate:

- **Store Location** – the location of the store where the mutual TLS certificate is located.
The default is `LocalMachine`.
- **Store Name** – the name of the store where the mutual TLS certificate is located. The default is `Personal`.
- **mTLS Thumbprint** – the thumbprint of the mutual TLS certificate.

For more information, including how to obtain the thumbprint of the certificate, see section [4.2.1, Mutual TLS](#).

Click **Next**, and the MyID Theme screen appears.



8. Type the display name that was provided for the Relying Party Trust for which the AD FS Adapter Mobile will provide the authentication.

To find the display name, look in the following location:

Server Manager > Tools > AD FS Management > AD FS > Relying Party Trusts > Display Name

For more information on themes, see section [4.4, Managing the AD FS Adapter Mobile](#).

Click **Next**, then click **Install**.

9. When the installation program has completed, click **Finish**.

4.3.1 Uninstalling the AD FS Adapter Mobile

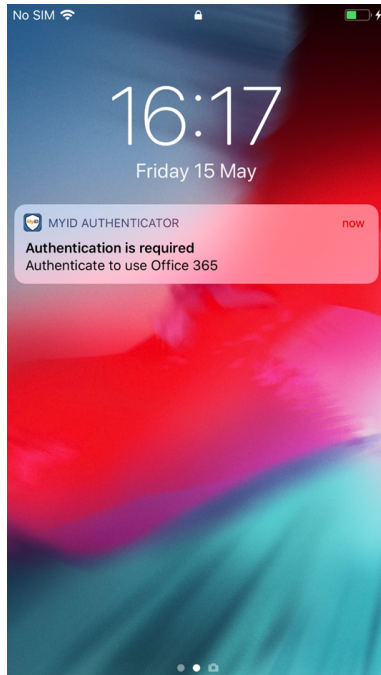
You can uninstall the AD FS Adapter Mobile from the **Apps & features** section of Windows Settings; it is listed as the **AD FS Adapter for MyID**.

Note: Uninstalling the AD FS Adapter Mobile also uninstalls the AD FS Adapter OAuth, if you have it installed.

4.3.2 Notification display text

The AD FS Adapter installation program allows you to specify the title and description for the notification that appears to the user on the mobile device when authentication is required.

For example, if you set the title to "Authentication is required" and the description to "Authenticate to use Office 365", the notification would appear on an iPhone as follows:



4.4 Managing the AD FS Adapter Mobile

After you have installed the AD FS Adapter Mobile, you can manage its settings using a provided suite of PowerShell scripts and a JSON configuration file.

4.4.1 Configuration file

The AD FS Adapter Mobile configuration is stored in a JSON file called `MobileAdfsAdapter.json` in the `ADFS_Adapter_Mobile` folder.

If you have manually unregistered the AD FS Adapter Mobile and want to register it again, you can run the following PowerShell script:

- `RegisterADFSProvider.ps1` – this script reads the information in the configuration file and uses it to register the AD FS Adapter Mobile.

You can also make changes to the configuration file and apply new settings.

To edit and apply new configuration settings:

1. In the `ADFS_Adapter_Mobile` folder, open the following file in a text editor:

`MobileAdfsAdapter.json`

2. Edit the following values:

- `logFilePath`
- `logLevel`

For details of setting up logging, see section [4.5.1, *Setting up AD FS Adapter Mobile logging*](#).

- `maxRetries` – set this to the number of polling attempts the AD FS Adapter will make to the MyID Verification Service to check if verification is complete before failing.

The default is 100.

- `retryDelayMs` – set this to the delay in milliseconds between each polling attempt retry.

The default is 1000.

The configuration file also contains settings that you provided when running the installation program, including:

- `server` – the URL of the MyID Verification Service.
- `userlookup` – the method of user identification with the MyID Verification Service. This can be one of the following:
 - `"incoming-claim"` – based on the claim provided by the Relying Party, which is predominantly UPN (this is the default).
 - `"upn"` – treats the input as an UPN.
 - `"emailaddress"` – treats the input as an email address.
- `requestBody` – the push notification settings, including:
 - `title` – the title of the notification.
 - `body` – the text in the notification.
- `certFinder` – the details of the mutual TLS certificate, including:
 - `storeLocation` – the location of the store.
 - `storeName` – the name of the store.
 - `mTlsThumbprint` – the thumbprint of the certificate.

You are recommended to use the installation program if you want to change any of these settings. See section [4.3, *Installing the AD FS Adapter Mobile*](#) for details.

3. Save the `MobileAdfsAdapter.json` file.

4. Run the `ReconfigureADFSPProvider.ps1` PowerShell script to apply the changes.

This script unregisters the AD FS Adapter, then re-registers it using the updated settings.

4.4.2 Managing themes

After you have installed the AD FS Adapter, the Intercede branding files are stored in the `Themes` folder in the installation folder.

Note: The themes folder is shared between the AD FS Adapter OAuth and the AD FS Adapter Mobile, if you have both installed.

The `MyIDAuthTheme2019` folder contains files used for Windows Server 2019 or Windows Server 2022, and includes custom images, CSS, JavaScript and HTML. You are not expected to edit these files. The `MyIDAuthTheme` folder contains files previously used for systems running Windows Server 2016.

You can apply and remove these themes using the following PowerShell scripts:

- `ApplyCustomTheme.ps1`
This script applies the Intercede branding to the Relying Party Trust selected at installation time.
- `RemoveCustomTheme.ps1`
This script removes the Intercede branding from the Relying Party Trust selected at installation time.

4.5 AD FS Adapter Mobile logging

The AD FS Adapter provides both its own logging and Windows Event logging.

4.5.1 Setting up AD FS Adapter Mobile logging

The AD FS Adapter Mobile log file has a rolling interval of a day, which means that a new file is created if needed each day.

Entries in the log file use the following template:

```
YYYY-MM-DD HH-MM-SS.mmm +UTC [Log Level shortened] (thread id) message
```

For example:

```
2020-05-12 09:16:30.786 +00:00 [INF] (10) mTLS Client <- Web REPLY:
{"status":"Success"}
```

You can change the location of the log file, and the logging event level.

To configure the logging settings:

1. In the `ADFS_Adapter_Mobile` folder, open the following file in a text editor:

```
MobileAdfsAdapter.json
```

This is a JSON file that contains the configuration settings for the AD FS Adapter Mobile . For more information on this file, see section [4.4.1, Configuration file](#).

2. Edit the following values:

- `logFilePath` – type the full path and name of the text file you want to use for logging. Use double backslashes in the path.

The date is automatically inserted before the file extension.

To switch off logging, set this to an empty string "". This is the default.

For example:

```
"C:\\Logs\\mobile-cust-auth-log-.txt"
```

This produces logs in the `C:\Logs\` folder with filenames similar to:

```
mobile-cust-auth-log-20201231.txt
```

Note: You must set up permissions on this folder so that the AD FS service account can write to this location.

- `LogLevel` – type the level of logging you want to occur.

From most logging to least logging, the levels you can use are:

- Verbose
- Debug
- Information
- Warning
- Error – this is the default level.
- Fatal

3. Save the `MobileAdfsAdapter.json` file.

4. Run the `ReconfigureADFSProvider.ps1` PowerShell script to apply the changes.

See section [4.4.1, Configuration file](#) for details.

4.5.2 Viewing the Windows Event log

When you have installed and configured the AD FS Adapter Mobile as an authentication method for a Relying Party Trust, when an authentication process begins, the AD FS Adapter Mobile raises a Windows application event showing the configuration loaded when AD FS started the AD FS Adapter Mobile. This shows the latest AD FS Adapter Mobile configuration provided by the installer or reconfiguration script.

If the AD FS Adapter Mobile encounters a problem, it raises a Windows application error event describing the problem.

To see these events, go to:

Event Viewer > Windows Logs > Application

Additionally, if the calling AD FS service detects a problem from the AD FS Adapter Mobile, it raises an error event in the following location describing the problem from the AD FS point of view.

Event Viewer > Applications and Service Logs > AD FS > Admin