
# MyID PIV

**Version 12.14**

# Passkey Integration Guide

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

Document reference: IMP2008-12.14.0-PIV

March 2025

# Copyright

# 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

# 1 Introduction

This guide provides details of how to integrate your MyID® system with passkeys using FIDO (Fast IDentity Online) authenticator devices.

Passkeys are a modern solution for multi-factor authentication. These credentials are cryptographically secure and provide an alternative to complex certificate-based solutions. The technology is supported directly by browsers and offers phishing-resistant strong authentication to websites and web applications.

Organizations that need to manage credentials for end users of their systems typically have much more complex requirements than consumers; for example, ensuring that the right people get the right credential, enforcing security policies, and the ability to track who has a credential are all vital. Critically, the ability to revoke access in the simplest and fastest way when required is essential.

To deploy passkeys in an enterprise, organizations need an authentication server to hold the registration information for the credential and to perform the authentication process when required. You can use MyID CMS as an authentication server to issue your passkeys, or you can use Microsoft Entra. Integration of multiple authentication servers can be challenging for organizations due to the additional complexity this brings, so using Entra ID as the primary identity provider while using MyID CMS for credential management brings together the best of both solutions.

Intercede also provides a plug-in for AD FS (the MyID AD FS Adapter OAuth) that allows you to use the MyID authentication service in conjunction with a registered passkey to access AD FS (Active Directory Federation Services); see the *MyID AD FS Adapter OAuth* section in the ***MyID Authentication Guide*** for details.

You can integrate MyID's authentication service with your own system to authenticate a person's identity using their passkey using OAuth 2.0 OpenID Connect; see the *Authenticating using OpenID* section in the ***MyID Authentication Guide*** for details.

You can also set up the MyID authentication service as a standalone service (for high availability passkey authentication operations); see the *Setting up the standalone authentication service* section in the ***MyID Authentication Guide*** for details.

## 1.1 Supported authenticators

MyID can work with any FIDO compatible authenticator that meets the technical standards set by the FIDO Alliance and that can pass device attestation checks.

**Note:** Platform authenticators are *not* supported for issuance through Entra.

Passkeys may provide single-factor, two-factor, or multi-factor authentication. You can configure MyID to treat FIDO basic assurance authenticators and high assurance authenticators with different levels of trust; for example, you can enable logon to MyID for high assurance authenticators, but disable logon for basic assurance authenticators.

See section *5*, *Setting up credential profiles for passkeys* and section *2.6*, *Configuring MyID for logon with passkeys* for details.

## 1.2 Supported browsers

The passkey registration process is supported on the following browsers:

- Google Chrome
- Microsoft Edge (Chromium version)
- Mozilla Firefox.

**Note:** The pre-Chromium version of Microsoft Edge is not supported.

# 2 Issuing passkeys through MyID CMS

You can use MyID CMS as the authentication server for issuing passkeys.

For information on configuring MyID to issue passkeys for use with Entra, see section *3*, *Issuing passkeys with Microsoft Entra*.

To configure your system to issue passkeys using MyID CMS for authentication, you must carry out the following:

- Set the MyID configuration options.

  See section *2.1*, *Setting the configuration options*.
- Set up your MyID system for email notifications.

  See section *2.2.1*, *Configuring your system for email notifications*.
- Configure your email templates for the FIDO issuance notification and the registration code.

  See section *2.2.2*, *Editing the notification templates*.
- Optionally, set up SMS for mobile notifications.

  See section *2.2.3*, *Configuring the registration code to be sent by SMS*.
- Configure the MyID authentication service with the FIDO metadata.

  See section *2.3*, *Setting up the FIDO metadata*.
- If necessary, amend the server settings.

  See section *2.4*, *Configuring the server settings*.
- Configure your end-user roles to allow for FIDO registration.

  See section *2.5*, *Configuring roles for registering passkeys*.
- Optionally, configure MyID for logon using a passkey.

  See section *2.6*, *Configuring MyID for logon with passkeys*.
- Check the IIS configuration.

  See section *2.7*, *Checking IIS configuration*.

## 2.1 Setting the configuration options

To set the configuration options required for registering passkeys:

1. Log on to MyID Desktop as an administrator.
2. From the **Configuration** category, select **Operation Settings**.
3. On the **General** tab, set the following option:
   - **URL path** – make sure this is set to the URL of the MyID web server. Include the protocol and server name only; for example:

     `https://myserver.example.com`

     MyID uses this option to generate the link to the registration page for passkeys on the authentication service.

     **Important:** Your MyID system must be set up to use https.
4. Click **Save changes**.
5. From the **Configuration** category, select **Security Settings**.
6. On the **Logon** tab, set the following option:
   - **Allow Logon Codes** – set this option to `Yes`.

     MyID uses this option to generate a single-use code for the passkey registration process.
7. Click **Save changes**.

## 2.2 Setting up email templates for passkey registration

When a person requests a passkey through the MyID Operator Client or the Self-Service Request Portal (assuming the credential profile is not set up for immediate registration), MyID sends two email messages; the first contains a link to the registration web page, and the second contains a single-use registration code.

You can edit the templates used for these messages; see section *2.2.2*, *Editing the notification templates*.

For increased security, you can configure MyID to send the registration code by SMS instead of email; see section *2.2.3*, *Configuring the registration code to be sent by SMS*.

### 2.2.1 Configuring your system for email notifications

You must configure your system with an SMTP server; see the *Setting up email* section in the ***Advanced Configuration Guide***.

You must also ensure that everyone who is going to request a passkey has an email address stored in their user account. You can use the **Requisite User Data** feature of the credential profile to prevent people who do not have email addresses from requesting passkeys; see section *5*, *Setting up credential profiles for passkeys*.

### 2.2.2 Editing the notification templates

To edit the content of the FIDO registration messages:

1. Log on to MyID Desktop as an administrator.
2. From the **Configuration** category, select **Email Templates**.
3. Select one of the following templates, and click **Modify**:
   - **Register passkey** – this template contains the link to the MyID authentication web page.

     You can provide the following information in this message:

     - `%x` – the URL of the MyID web server. This is taken from the URL Path configuration option; see section *2.1*, *Setting the configuration options*.
     - `%jobguid` – the ID of the request for the passkey.

       **Important:** The registration URL you provide in the message *must* have the following format:

       ```
       %x/web.oauth2/fido/register/begin?requestId=%jobguid
       ```
   - **FIDO Authenticator Registration Code** – this template contains the single-use registration code that the end user will enter to complete their registration.

     You can provide the following information in this message:

     - `%logonName` – the person's logon name.
     - `%logonCode` – the FIDO registration code.

### 2.2.3 Configuring the registration code to be sent by SMS

To allow MyID to send SMS messages, set the **SMS email notifications** on the **General** tab of the **Operation Settings** workflow to `Yes`.

By default, SMS messages are sent to an Email to SMS gateway, in the format `<cellnumber>@<gateway>`, where:

- `<cellnumber>` – the cell phone number from the user's record.
- `<gateway>` – the URL from the **SMS gateway URL for notifications** option on the **General** tab of the **Operation Settings** workflow.

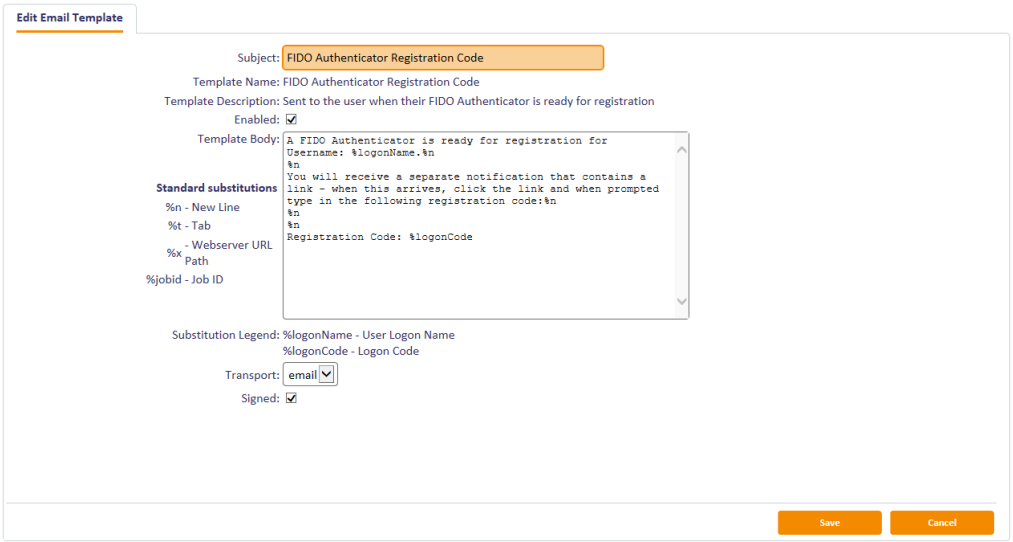
For example: `00447700900123@smsgateway.com`

If this is not suitable, you can customize the `sp_CustomPrepareSMS` stored procedure in the MyID database.

You must also edit the email template containing the registration code, and change it to use SMS instead of email for its transport:

1. Log on to MyID Desktop as an administrator.
2. From the **Configuration** category, select **Email Templates**.
3. Select the **FIDO Authenticator Registration Code** template, and click **Modify**.

   The Edit Email Template screen appears.




4. From the **Transport** drop-down list, select **sms**.
5. Click **Save**.

**Important:** If you configure the registration codes to be sent by SMS, you must ensure that everyone who is going to request a passkey has a mobile phone number stored in their user account. You can use the Requisite User Data feature of the credential profile to prevent people who do not have mobile phone numbers from requesting passkeys; see section *5*, *Setting up credential profiles for passkeys*.

## 2.3 Setting up the FIDO metadata

To allow MyID to carry out attestation checks on passkeys during registration, it requires access to the FIDO Alliance Metadata Service metadata. You can obtain FIDO metadata online (this is the default) or from a local file system repository.

Access to online metadata is controlled from the configuration file for the MyID authentication web service (web.oauth2) using the `OnlineMetadata` Fido setting:

```
"Fido": {
"Config": {
// If OnlineMetadata is true FIDO metatdata will be downloaded from
https://mds.fidoalliance.org/
"OnlineMetadata": true,
// Optionally provide a FIDO metadata Blob root certificate in base64 format
"MetadataBlobRootCert": "",
// Control whether the certificates in the chain of trust for the FIDO metadata service
BLOB is CRL checked.
"DisableCrlCheck": false,
}
}
```

The installer sets `OnlineMetadata` to `true` by default in the `appsettings.json` file, meaning that FIDO metadata is obtained from the online source. If you need to change any settings, you can edit the `appsettings.Production.json` file to override the settings in the `appsettings.json` file.

The MyID FIDO implementation uses an built-in FIDO Alliance ROOT certificate which is part of the certificate chain of trust for the signed metadata `blob.jwt` (see section *2.3.1*, *Setting up a local metadata repository*). This certificate has an expiry date of 2029. If this certificate expires, or the FIDO Alliance certificate chain changes, you can provide an external root certificate as a base64 string through the `MetadataBlobRootCert` setting above. The root certificate is then used to verify cryptographically any `blob.jwt` obtained online or used by the local metadata repositories.

For online metadata during FIDO registration, you can disable CRL checking of the metadata chain of trust certificates by setting `DisableCrlCheck` to `true`. The default value is `false` which means that CRL checking is enabled by default. The `DisableCrlCheck` setting applies only to online metadata; no CRL checking is done on a local metadata repository.

Previous versions of FIDO on MyID used the FIDO Alliance MDS2 metadata service, which required an access token to be able to download the metadata. With the newer MDS3 FIDO Alliance metadata service, this is no longer the case, with metadata now freely available as a direct download from:

***mds.fidoalliance.org***

The MyID web server must be able to access the above FIDO Alliance URL to download the metadata. If your server cannot access this URL, if you are experiencing performance issues when verifying metadata, or if you want to use metadata for authenticators not included in the public repository, you can create a local repository; see section *2.3.1*, *Setting up a local metadata repository*. You can also create a local repository to provide a restricted list of devices that can pass the attestation check.

**Note:** If you are using the standalone authentication service (web.oauth2.ext) in conjunction with the AD FS Adapter OAuth to allow for FIDO authentication to your AD FS, the web.oauth2.ext service configuration for `OnlineMetadata` is ignored, because attestation checks are relevant only for registration, and the standalone authentication service provides only authentication and not registration for passkeys.

### 2.3.1 Setting up a local metadata repository

You can create a local metadata repository, which you can then configure the authentication service to use in addition to or instead of the live metadata from the FIDO Alliance website.

You can set up a local repository for basic attestation or for enterprise attestation:

- For basic attestation, you can either use the MDS3 FIDO Alliance metadata or set up a custom repository that contains only those specific devices you want to issue.
- For enterprise attestation, you can set up a custom repository containing metadata for your enterprise attestation certificates.

When issuing passkeys, the attestation check used is determined by the **Require Attestation** option in the credential profile. See section *5.1*, *Setting up a passkey credential profile for the MyID Operator Client* or section *5.2*, *Setting up a passkey credential profile for the Self-Service Request Portal* for details.

To create a local repository:

1. Obtain your metadata repository.

   For example, you can download the MDS3 FIDO Alliance metadata; in your browser, navigate to:

   ***mds.fidoalliance.org***

   This downloads a single `blob.jwt` file that contains, amongst other things, the FIDO Alliance metadata statements for all registered passkeys.

2. Open the `appsettings.Production.json` file for the authentication service in a text editor.

   By default, this is:

   `C:\Program Files\Intercede\MyID\web.oauth2\appsettings.Production.json`

3. Edit the file to include the following settings:

```
"Fido": {
"Config": {
// If OnlineMetadata is true FIDO metatdata will be downloaded from
https://mds.fidoalliance.org/
"OnlineMetadata": false,
// If a filesystem FIDO metadata repository is provided by the customer
its path should be entered here.
"MDSCacheDirPath": "<path of basic metadata folder>",
// If a filesystem FIDO metadata repository is provided by the customer
and the customer wishes to override
// the nextUpdate time in the metadata BLOB file then this can be done by
setting the cache time in days from
// the time now.  A default value of 2 days ensures that during fido
registration the file is read only once
// from the file system with all subsequent times read from memory cache
until the system is restarted.
"MDSCacheDirPathEnterprise": "<path of enterprise attestation metadata
folder>",
"CacheTimeDays": <validity period>
}
}
```

   where:

- `<path of basic metatdata folder>` is the path of the folder where the `blob.jwt` is located. Use forward slashes or double backslashes in the path. A non-empty value here enables the local file system metadata repository.

  This folder must only ever contain metadata with root certificates that validate direct attestation certificates for a given AAGUID.

- `<path of enterprise attestation metatdata folder>` is the path of the folder where the `.jwt` or JSON files containing the enterprise attestation metadata is stored. Use forward slashes or double backslashes in the path. A non-empty value here enables the local file system enterprise attestation metadata repository.

  This folder must only ever contain metadata with root certificates that validate enterprise attestation certificates for a given AAGUID.

  See section *4*, *Enterprise attestation* for details of enterprise attestation checks.

- `<validity period>` is the number of days from the current time after which the cache will no longer be valid, and the authentication service will revert to re-reading the metadata from the file system rather than its own in-memory cache.

The installer sets a default value of 2 days for `CacheTimeDays` in the `appsettings.json` file. This means, after reading the file system metadata into memory cache, it will keep using the memory cache for two days (or until the server app pool is recycled) then re-read the file system metadata into memory cache and then use memory cache for another 2 days and so on. If you set the `CacheTimeDays` value to 0 it will be ignored and the `nextUpdate` time specified in the `blob.jwt` file will apply; when that expires it will keep reading from the file system.

**Important:** Merge these settings into your existing `Fido:Config` section. Do not delete any existing settings.

If you want to use the local repository in preference to the live data, you can set the `OnlineMetadata` option to `false` as shown above.

If you leave the `OnlineMetadata` set to `true` in the configuration file, MyID uses both live metadata and metadata from the local repository. You can use either or both options.

4. Save the `appsettings.Production.json` file.
5. Recycle the web service app pool:
   a. On the MyID web server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the myid.web.oauth2.pool application pool, then from the pop-up menu click **Recycle**.

This ensures that the web service has picked up the changes to the configuration file.

**Note:** Earlier versions of FIDO on MyID used a `MyID.FIDO.Metadata.App` to obtain a local metadata repository from the MDS2 server. With MDS3 no such app is required as the metadata is all in a single `blob.jwt` file that is freely available from the FIDO Alliance MDS3 server.

Existing MDS2 metadata JSON files are not compatible with the MDS3 metadata specification. This means customers who already have MDS2 metadata JSON files in their local repository must replace them with a single MDS3 compatible `blob.jwt` metadata file.

If you place individual MDS3-compliant JSON metadata statement files alongside the `blob.jwt` in the file system repository folder and they are read in and appended to the metadata extracted from the `blob.jwt` file. This is useful if you want to use metadata for authenticators that have not been registered with the FIDO Alliance.

Alternatively, you can populate your metadata folder with individual MDS3-compliant JSON metadata statement files *instead of* the `blob.jwt`, in which case the data in these files is used instead of the general metadata repository; this allows you to support specific tokens.

JWT metadata files are cryptographically verified as originating from the FIDO Alliance before being accepted as a source of metadata.

Only one JWT file is supported in the local repository path. If more than one file is present, only the first found alphabetically will be used. However, you can include many individual JSON metadata files in the local repository path.

#### 2.3.1.1 Metadata repository priority

When MyID CMS reads the AAGUID (a 128-bit identifier of the type of the authenticator) from the authenticator device, it tries to match the identifier against the metadata repositories in the following priority order:

- The local enterprise attestation repository, as configured using the `MDSCacheDirPathEnterprise` option.
- The local basic attestation repository, as configured using the `MDSCacheDirPath` option.
- The online MDS3 FIDO Alliance metadata, as configured using the `OnlineMetadata` option.

As soon as MyID CMS finds a match for the AAGUID, it does not proceed to look any further; this means that you cannot have multiple sets of metadata for a single AAGUID, as MyID CMS always uses the first match.

## 2.4 Configuring the server settings

This section contains information on configuring the server settings in the `appsettings.Production.json` file.

### 2.4.1 FIDO configuration file options

The server settings are derived from the value you provided for the **MyID Server URL** in the MyID installation program, and are initially stored in the `appsettings.json` file. If you need to change these settings, you can edit the `appsettings.Production.json` file to override the settings in the `appsettings.json` file.

**Note:** If you subsequently install or upgrade MyID again and provide a different value in the **MyID Server URL** in the MyID installation program, and you have set the `Origin`, `Origins`, or `ServerDomain` options in the `appsettings.Production.json` file, the values you enter in the installation program are ignored; the `appsettings.Production.json` file is never updated by the installation program, and always takes precedence over the `appsettings.json` file.

- `ServerDomain` is used when registering the FIDO to instruct the FIDO token the domain that may be used during authentication. It must be either:
  - The exact web domain name that will be used during authentication. In this case, only this domain may be used during authentication, *or*:
  - A registrable domain suffix of the web domain that will be used during authentication. In this FIDO authentication will be possible either using this exact domain, or any sub-domain of that domain
- `Origin` or `Origins` specify the origins that are allowable to be used during FIDO authentication.

  Use `Origin` if you have a single origin, or `Origins` if you have multiple origins; if you specify a value for `Origins`, it overrides any setting you provide for `Origin`.

For example: web.oauth2 is running on `https://myid.customer.com`

In this case, set `Origin` to `https://myid.customer.com` and `ServerDomain` to `myid.customer.com` – this allows tokens to be registered on the `myid.customer.com` domain so that they can authenticate only on `myid.customer.com`.

Alternatively: you intend to register FIDO tokens in web.oauth2 on `https://customer.com` and for those tokens to authenticate to that instance of MyID, but that instance of MyID is also reachable through a sub-domain `https://subdomain.customer.com`

In this case, on web.oauth2 set the `Origins` to:

```
["https://customer.com","https://subdomain.customer.com"]
```

This allows authentication on either of those origins. Then set `ServerDomain` to:

```
customer.com
```

That is, the registrable domain suffix, which means that the FIDO token can be used to authenticate on the `customer.com` domain or any sub-domain of `customer.com`, subject to the `Origin` also being listed in the `Origin` or `Origins` section.

### 2.4.2 Single origin

To configure the server settings for a single origin:

1. As an administrator, open the `appsettings.Production.json` file in a text editor.

   By default, this is:

   `C:\Program Files\Intercede\MyID\web.oauth2\appsettings.Production.json`

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

2. Edit the file to include the following:

```
{
    "Fido":{
        "Config":{
            "Origin":"https://<server>:<port>",
            "ServerDomain":"<server>"
        }
    }
}
```

   where:

   - `<server>` – the name of the server to which users will authenticate.
   - `<port>` – optionally, the port to which users will authenticate, if you are using a non-standard HTTPS port.

   You must add the `Origin` and `ServerDomain` to any existing entries in the `Fido:Config` section. Your `appsettings.Production.json` file may already contain commented-out entries for these values; remove the double-slash `//` to uncomment the entries.

   **Important:** The `Origin` and `ServerDomain` options are case sensitive, and must be consistent with the casing of the DNS Name in the web server's TLS certificate.

   For example:

```
{
  "Fido":{
    "Config":{
      "Origin": "https://myserver.example.com:30443",
      "ServerDomain": "myserver.example.com"
    }
  }
}
```

3. Save the `appsettings.Production.json` file.

4. Recycle the web service app pool:
   a. On the MyID web server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **myid.web.oauth2.pool** application pool, then from the pop-up menu click **Recycle**.

   This ensures that the web service has picked up the changes to the configuration file.

### 2.4.3 Multiple origins

MyID has support for multiple origins, where multiple sub-domains of a registrable domain can be used for authentication.

To configure the server settings for multiple origins:

1. As an administrator, open the `appsettings.Production.json` file in a text editor.

   By default, this is:

   `C:\Program Files\Intercede\MyID\web.oauth2\appsettings.Production.json`

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

2. Edit the file to include the following:

```
{
  "Fido":{
    "Config":{
      "Origins":["https://<server>:<port>",
"https://<subdomain1>:<port>", "https://<subdomain2>:<port>" ... ],
      "ServerDomain":"<server>"
    }
  }
}
```

   where:

   - `<server>` – the name of the server that contains the sub-domains to which users will authenticate.
   - `<subdomain<i>x</i>>` – A list of sub-domains of the server domain that will be allowed to authenticate.
   - `<port>` – optionally, the port to which users will authenticate, if you are using a non-standard HTTPS port.

   **Important:** The `Origins` and `ServerDomain` options are case sensitive, and must be consistent with the casing of the DNS Name in the web server's TLS certificate.

3. Save the `appsettings.Production.json` file.
4. Recycle the web service app pool:
   a. On the MyID web server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **myid.web.oauth2.pool** application pool, then from the pop-up menu click **Recycle**.

   This ensures that the web service has picked up the changes to the configuration file.

**Note:** If `Origins` is specified then it overrides any value in `Origin`.

## 2.5 Configuring roles for registering passkeys

Any person who wants to register a passkey must have a role that has permission to use the **Register FIDO Security Key** option.

To configure a role for registering passkeys:

1. Log on to MyID Desktop as an administrator.
2. From the **Configuration** category, select **Edit Roles**.
3. Click **Show/Hide Roles** to display the role to which you want to add the FIDO registration permission.

   **Note:** This role *must* have access to the Password logon mechanism; the FIDO registration code is a special case of a logon code, and logon codes use the Password logon mechanism.
4. From the **Cards** section, select the following option:
   - **Register FIDO Security Key**

   **Note:** If you are using the Self-Service Request Portal to request and register passkeys, you must set up the **Derived Credential Owner** role to have access to the Password logon mechanism and the **Register FIDO Security Key** option.
5. Click **Save Changes**.

Any person who has the selected role can now access the authentication service to register a passkey.

## 2.6 Configuring MyID for logon with passkeys

If you want to allow people to log on to your MyID system using their registered passkeys, you can configure MyID to allow FIDO logon.

Configuring FIDO logon requires the following:

- Setting up global configuration options for FIDO logon.
- Configuring individual roles for FIDO logon.

**Note:** You are recommended to restrict logon to multi-factor authenticators; MyID allows you to differentiate between Basic assurance authenticators and High assurance authenticators. If you allow logon for Basic assurance authenticators, you are recommended to allow those users access only to read-only features with limited scope.

### 2.6.1 Setting the FIDO logon configuration options

You can enable or disable FIDO logon globally using the configuration options.

To enable or disable FIDO logon:

1. Log on to MyID Desktop as an administrator.
2. From the **Configuration** category, select **Security Settings**.
3. On the **Logon Mechanisms** tab, set the following options:
   - **FIDO Basic Assurance Logon** – set this option to `Yes` to enable logon to MyID with a passkey that has been issued with a credential profile where the **Assurance Level** is set to **Basic**.
   - **FIDO High Assurance Logon** – set this option to `Yes` to enable logon to MyID with a passkey that has been issued with a credential profile where the **Assurance Level** is set to **High**.
4. Click **Save changes**.

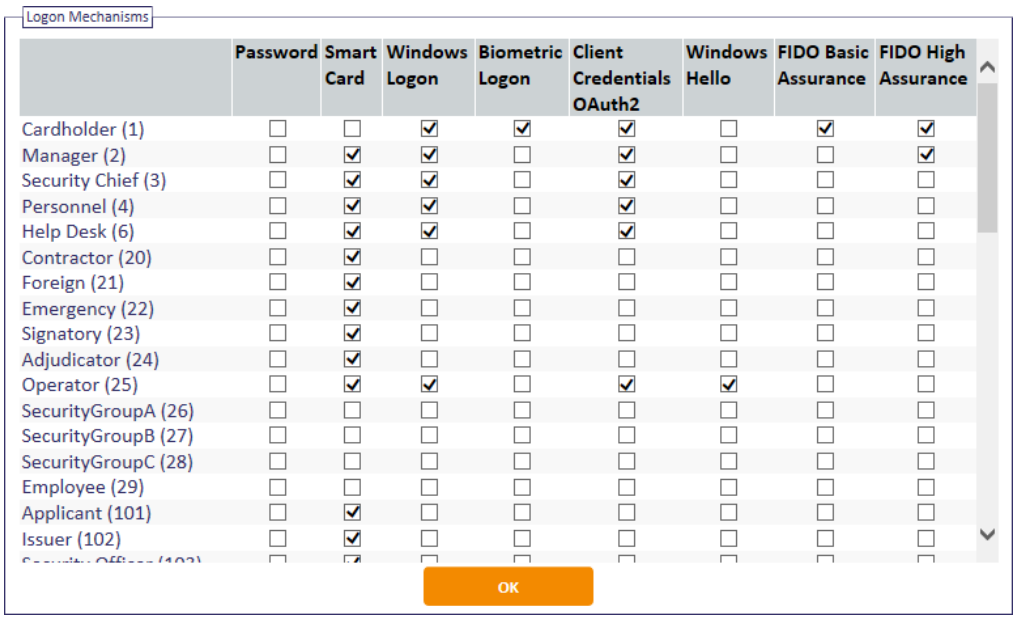
## 2.6.2 Setting up FIDO logon mechanisms

For each role in MyID, you can decide whether people who have been assigned that role can log on to MyID using their registered passkey and access the features that are configured for that role.

**Note:** If a person has multiple roles, but has a FIDO logon mechanism configured for only some of them, when they log on to MyID using their passkey they can access only those features that are configured for the roles that have the FIDO logon mechanism configured. For example, if Susan has a Cardholder role with access to **View Person**, and Reporter role with access to **Management Information Reports**, if only the Cardholder role has a FIDO logon mechanism configured, when she logs on to MyID using her passkey, she can access only View Person; to access her reports, she must log on with a smart card.

1. Log on to MyID Desktop as an administrator.
2. From the **Configuration** category, select **Edit Roles**.
3. Click **Logon Methods**.

   The Logon Mechanisms screen appears.




4. For each role you want to be able to log on to MyID using FIDO, select one of the following options:
   - **FIDO Basic Assurance** – access to the features configured for this role is allowed when logging on with a passkey that has been issued with a credential profile where the **Assurance Level** is set to **Basic**.
   - **FIDO High Assurance** – access to the features configured for this role is allowed when logging on with a passkey that has been issued with a credential profile where the **Assurance Level** is set to **High**.
5. Click **OK**.
6. Click **Save Changes**.

MyID is now configured for logon using passkeys. For information on using passkeys to log on, see section *6.5*, *Signing in to MyID CMS with a passkey*.

## 2.7 Checking IIS configuration

You must make sure that your Internet Information Services (IIS) is set up with the correct feature delegation options.

In Internet Information Services (IIS) Manager, carry out the following:

1. In the **Connections** pane, select the web server.
2. In **Features View**, in the **Management** section, double-click the **Feature Delegation** option.
3. Ensure that the following are set:
   - **Authentication - Anonymous** – make sure this is set to **Read/Write**.
   - **Authentication - Windows** – make sure this is set to **Read/Write**.

   To change the setting, right-click the option, then from the pop-up menu select **Read/Write**.

If your system has these options set to **Read Only**, you may experience a problem using FIDO to authenticate, with a `500 30` server error message.

# 3 Issuing passkeys with Microsoft Entra

You can use Microsoft Entra as the authentication server for issuing passkeys, while still using MyID CMS for credential management.

Within Entra, you require an Enterprise Application registered to allow the MyID server to access the APIs. You require information about this application when you set up the external system that allows MyID to communicate with the Entra server; see section *3.7.3*, *Setting up the external system*.

A primary requirement for MyID to issue passkeys with Microsoft Entra is that there is a link between the user's Entra account and their MyID account.

One way to achieve this is to add the `ObjectGUID` of the user's account in Entra to the MyID CMS database. Currently, the user's Entra ID `ObjectGUID` is not typically available in an on-premise Active Directory, so you cannot synchronize it into the MyID database automatically.

This means that to add the Entra `ObjectGUID` to MyID, you must use one of the following processes:

- Register the passkey using the MyID Self-Service Request Portal configured to use Microsoft Entra for authentication.

  See section *3.1*, *Registering the Entra ID using the Self-Service Request Portal*.

  This is the recommended method for this release, as the Entra `ObjectGUID` is added to the MyID database automatically when you use Entra for authentication to the Self-Service Request Portal. If you want to use an alternative method of issuing the passkey, you must add the `ObjectGUID` to the person's MyID account first.
- Add the Entra `ObjectGUID` to the user's MyID account using the MyID Operator Client.

  See section *3.2*, *Updating the Entra ID using the MyID Operator Client*.
- Add the Entra `ObjectGUID` to the user's MyID account using the MyID Core API.

  See section *3.3*, *Updating the Entra ID using the MyID Core API*.
- Add the Entra `ObjectGUID` to the user's MyID account by updating the MyID database directly.

  See section *3.4*, *Updating the Entra ID in the database directly*.

Alternatively, MyID also supports the use of the User Principal Name as the matching criteria to Entra ID; in this case, as the User Principal Name may already be known to your MyID system through Active Directory synchronization, you do not need to add the Entra `ObjectGUID` manually or through the Self-Service Request Portal.

Once you have configured MyID to link the user's Entra account to the their MyID account, you can issue passkeys using a variety of standard MyID issuance processes, including:

- Creating an issuance request in MyID, with MyID authentication as an alternative to Entra authentication during the registration process.
- Using smart card logon as authentication to collect the passkey.
- Through the Self-Service Request Portal with certificate-based authentication (derived credentials). When you use this method, if the certificate used for authentication has been issued by a different system, the UPN of the user account in Entra must be present in the certificate that initiates the request.

You can view a person's Entra ID on the **Account** tab of the **View Person** screen in the MyID Operator Client; by default, the Entra ID is stored in the **External Reference ID 1** field, but you can also use the **External Reference ID 2** and **External Reference ID 3** fields.

**Note:** The registration process currently uses a combination of the Self-Service Request Portal web page and installed MyID client software (MyID Client Services) – however, at this time, collection using the MyID Self-Service App or MyID Desktop is not supported.

You must carry out the following to configure your system to issue passkeys with Microsoft Entra:

- Configure your Microsoft Entra system.

  See section *3.5*, *Configuring Microsoft Entra*.
- Configure the Self-Service Request Portal.

  See section *3.6*, *Configuring the Self-Service Request Portal*.
- Configure your MyID system.

  See section *3.7*, *Configuring MyID to issue passkeys using Microsoft Entra*.

## 3.1 Registering the Entra ID using the Self-Service Request Portal

You must configure the Self-Service Request Portal to use Microsoft Entra as an external identity provider. See section *3.5.1*, *Configuring Entra to allow access from MyID CMS* for details.

The process for issuing a passkey with Microsoft Entra using the Self-Service Request Portal is as follows:

1. In Entra, the administrator sets up authentication for the user.

   You can use any form on Entra authentication; for example, you can create a Temporary Access Pass (TAP) for the end user, set them up with a password or certificate-based authentication, or set up pass-through authentication using the user's Windows login.

   The requirement is that the user can authenticate to Entra during the passkey registration process.
2. The administrator provides the user a link to the MyID Self-Service Request Portal.
3. The end user follows the link to the Self-Service Request Portal.
4. The end user authenticates to Entra.

   **Note:** If you are using pass-through authentication for your Entra authentication, this is seamless – the user does not have to provide any details, as their Windows account authenticates them automatically to Entra.

   During this process, MyID establishes a link between the MyID user account and the Entra ID account. This may involve importing the user details from Entra into MyID, or linking the Entra user to an existing MyID user.
5. The end user enrolls the passkey, which is then registered in Entra ID and in MyID CMS.

   The end user can then use the passkey for authentication to Entra ID, subject to Entra policies for authentication.

For more details on configuring your system to use the Self-Service Request Portal for issuing passkeys, see section *3.6*, *Configuring the Self-Service Request Portal*.

## 3.2 Updating the Entra ID using the MyID Operator Client

You can update the field that contains the Entra `ObjectGUID` using the Edit Person or Edit PIV Applicant screen in the MyID Operator Client.

To update the Entra ID for a person:

1. Search for a person, and view their details.

   See the *Searching for a person* section in the ***MyID Operator Client*** guide.

2. Click the **Edit Person** or **Edit PIV Applicant** option in the button bar at the bottom of the screen.

   You may have to click the **...** option to see any additional available actions.

3. Click the **Account** tab.

   **Note:** If the person's account is linked to a directory, you cannot edit the settings on this tab unless you have set the **Edit Directory Information** option on the **LDAP** tab of the **Operation Settings** workflow to `Yes`.

   Set one of the following options:

   - **External Reference ID 1** – maps to the `XuSYSExternalReferenceId1` field in the MyID database. This is the default field to store the Entra ID.
   - **External Reference ID 2** – maps to the `XuSYSExternalReferenceId2` field in the MyID database.
   - **External Reference ID 3** – maps to the `XuSYSExternalReferenceId3` field in the MyID database.
   - **User Principal Name** – maps to the `UserPrincipalName` field in the MyID database.

   **Note:** If you use a different field than `XuSYSExternalReferenceId1`, you must make sure that you configure the **External Entra Reference** option in the **External Systems** workflow to use the appropriate field; see section *3.7.3*, *Setting up the external system*.

4. Click **Save**.

## 3.3 Updating the Entra ID using the MyID Core API

You can update a person's user record in MyID with their Entra `ObjectGUID` using the MyID Core API.

Assumes that the person's MyID `ObjectID` is:

`A488BFA9-1460-4638-8C22-00DAB3B0F2BC`

and that their Entra `ObjectGUID` is:

`C47486E7-DDFE-43A6-9954-BDD1DF6AA743`

You can use the following endpoint to update the details for a person:

- `PATCH /api/People/`

By default, MyID uses the `XuSYSExternalReferenceId1` field in the `vPeopleUserAccounts` view to store the person's Entra `ObjectGUID`.

**Note:** If you use a different field than `XuSYSExternalReferenceId1`, you must make sure that you configure the **External Entra Reference** option in the **External Systems** workflow to use the appropriate field; see section *3.7.3, Setting up the external system*.

To specify this field through the API, use the following payload:

```
{
  "externalReferences": {
    "id1": "<entraID>"
  }
}
```

where `<entraID>` is the Entra `ObjectGUID` you want to add to the user's record.

For more information about using the MyID Core API, see the ***MyID Core API*** guide.

Once you have linked the person in MyID to the account in Entra, you can request a FIDO device using a credential profile configured for Entra.

The following examples assume your server is on `myserver.example.com`, and that you have already obtained an access token; `<YOUR-TOKEN>` is used as a placeholder.

### 3.3.1 cURL

```
curl.exe -X "PATCH" "https://myserver.example.com/rest.core/api/People/A488BFA9-1460-4638-
8C22-00DAB3B0F2BC?confirm=false" -H "Authorization: Bearer <YOUR TOKEN>" -H "accept:
application/json" -H "x-api-version: 1" -H "Content-Type: application/json" -d "
{""externalReferences"": {""id1"" : ""C47486E7-DDFE-43A6-9954-BDD1DF6AA743""}
```

**Note:** As in the MyID Operator Client, the standard Edit Person operation is not permitted for PIV applicants. To specify the Edit PIV Applicant operation instead, add `&op=200103` to the URL parameters; for example:

```
curl.exe -X "PATCH" "https://myserver.example.com/rest.core/api/People/A488BFA9-1460-4638-
8C22-00DAB3B0F2BC?confirm=false&op=200103" -H "Authorization: Bearer <YOUR TOKEN>" -H
"accept: application/json" -H "x-api-version: 1" -H "Content-Type: application/json" -d "
{""externalReferences"": {""id1"" : ""C47486E7-DDFE-43A6-9954-BDD1DF6AA743""}}"
```

### 3.3.2 Python

```
import requests
import json

# Set the server
server = "myserver.example.com"

# Set the ID of the person in the MyID database
personID = "A488BFA9-1460-4638-8C22-00DAB3B0F2BC"

# Set the ID of the person in Entra
entraID = "C47486E7-DDFE-43A6-9954-BDD1DF6AA743"

# Set the access token
token = "<YOUR TOKEN>"

# Build the payload
personData = {
    "externalReferences": {
        "id1": entraID
    }
}

person = json.dumps(personData)

# Call the API
response = requests.patch(
    "https://" + server + "/rest.core/api/People/" + personID + "?confirm=false",
    headers={"Authorization": "Bearer " + token,
             "Content-Type": "application/json",
             "accept": "application/json"},
    data=person)

# Display the response
if response.status_code==200:
    returnedData = json.loads(response.text)
    print(returnedData)
else:
    print("An error occurred:")
    returnedData = json.loads(response.text)
    print("Error code: " + returnedData["code"])
    print("Error message: " + returnedData["message"])
```

**Note:** As in the MyID Operator Client, the standard Edit Person operation is not permitted for PIV applicants. To specify the Edit PIV Applicant operation instead, add `&op=200103` to the URL parameters; for example:

```
"https://" + server + "/rest.core/api/People/" + personID + "?confirm=false&op=200103",
```

### 3.3.3 PowerShell

```
# Set the server
$server = "myserver.example.com"

# Set the ID of the person in the MyID database
$personID = "A488BFA9-1460-4638-8C22-00DAB3B0F2BC"

# Set the ID of the person in Entra
$entraID = "C47486E7-DDFE-43A6-9954-BDD1DF6AA743"

# Get the access token
$token = "<YOUR TOKEN>"

# Build the payload
$certData = "{'externalReferences': {'id1' : '" + $entraID + "'}}"

# Set up the call for the API
$authHeader = @{
    'Content-Type'='application/json'
    'Authorization'="Bearer $token"
    'x-api-version'= '1'
 }
$URI = 'https://' + $server + '/rest.core/api/People/' + $personID + '?confirm=false'
$person  = @{
    Headers =  $authHeader
    Uri = $URI
    Method = "PATCH"
    Body = $certData
}

# Display the response
try {
    $result = Invoke-WebRequest @person | ConvertFrom-Json
    Write-Host $result
}
catch {
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host "An error occurred:"
    Write-Host "Error code:" $responseBody.code
    Write-Host "Error message:" $responseBody.message
}
```

**Note:** As in the MyID Operator Client, the standard Edit Person operation is not permitted for PIV applicants. To specify the Edit PIV Applicant operation instead, add `&op=200103` to the URL parameters; for example:

```
$URI = 'https://' + $server + '/rest.core/api/People/' + $personID +
'?confirm=false&op=200103'
```

## 3.4 Updating the Entra ID in the database directly

**Important:** Back up your database before making any manual changes.

To update the database directly, you must know the `ObjectID` for the person in the MyID database and the `ObjectGUID` for the person in Entra.

The person's `ObjectID` is available in the `vPeopleUserAccounts` view in the MyID database. You can also see this ID in the URL of the View Person screen in the MyID Operator Client; for example:

`https://myserver.example.com/MyID/OperatorClient/#/people/A488BFA9-1460-4638-8C22-00DAB3B0F2BC`

This example assumes that the person's MyID `ObjectID` is:

`A488BFA9-1460-4638-8C22-00DAB3B0F2BC`

and that their Entra `ObjectGUID` is:

`C47486E7-DDFE-43A6-9954-BDD1DF6AA743`

To add the Entra ID to the `XuSYSExternalReferenceId1` field for the person's user account in MyID, run the following SQL statement against the MyID database:

```
update vPeopleUserAccounts set XuSYSExternalReferenceId1 = 'C47486E7-DDFE-43A6-9954-
BDD1DF6AA743' where ObjectID = 'A488BFA9-1460-4638-8C22-00DAB3B0F2BC'
```

**Note:** If you use a different field than `XuSYSExternalReferenceId1`, you must make sure that you configure the **External Entra Reference** option in the **External Systems** workflow to use the appropriate field; see section *3.7.3, Setting up the external system*.

Once you have linked the person in MyID to the account in Entra, you can request a FIDO device using a credential profile configured for Entra.

## 3.5 Configuring Microsoft Entra

You must configure your Microsoft Entra system to allow MyID to use it to issue passkeys.

If you are using the Self-Service Request Portal, you must also configure Microsoft Entra as an external identity provider for the SSRP.

### 3.5.1 Configuring Entra to allow access from MyID CMS

You must configure Entra to allow access from the MyID server:

- In your Entra tenant, use the **Create your own application** option to create an enterprise application; you can configure this to permit access from the MyID server to Entra.
- Set up an associated app registration for the enterprise application.
- You must create a client secret in Entra for server authentication from MyID.

  You must specify this client secret in MyID when you set up the external system; see section *3.7.3*, *Setting up the external system*.

  You must also specify the client secret when you configure the SSRP to use Microsoft Entra as an external identity provider.
- Make sure the Passkey (FIDO2) authentication method is enabled on your Entra system.

See the *External identity providers* section in the ***Derived Credentials Self-Service Request Portal*** guide on configuring Microsoft Entra as an external identity provider for the Self-Service Request Portal. In particular, you must:

- Configure the redirect URI to allow responses to be returned to SSRP.

  **Note:** You must configure the redirect URI as a web page, not a single-page application.

### 3.5.2 Configuring Entra to allow MyID to access the passkey registration APIs

MyID currently uses the following API methods:

- `GET /users/<UserID>/authentication/fido2methods/creationOptions`

  Retrieves the available options for creating a passkey.
- `POST /users/<UserID>/authentication/fido2methods`

  Registers a passkey.
- `DELETE /users/<UserID>/authentication/fido2Methods/<DeviceID>`

  Cancels a passkey.

You must grant the following Graph API permission to the app registration used by MyID:

| API / Permissions Name | Type | Description |
|---|---|---|
| `UserAuthenticationMethod.ReadWrite.All` | Application | Read and write all users' authentication methods |

### 3.5.3 Configuring Entra for enterprise attestation

When issuing a passkey through Entra, MyID CMS passes the enterprise attestation information to Entra; however, if you are using enterprise attestation for your passkey issuance that uses a custom attestation root certificate that is not publicly available in the FIDO repository, Entra is unable to carry out the attestation check. In this case, you must disable the attestation check on Entra.

MyID CMS carries out the enterprise attestation check on behalf of Entra; see section *2.3.1*, *Setting up a local metadata repository* for details of setting up a metadata repository that contains your custom attestation root certificate.

### 3.5.4 Restricting the number of credentials issued to a person

Entra uses a setting called `ExcludeCredentials` that prevents a person from collecting a second FIDO credential to the same FIDO device.

MyID CMS ignores this restriction and by default allows people to collect multiple FIDO credentials to the same FIDO device.

If you want to restrict the credentials issued through MyID CMS, you can set the **Active credential profiles per person** option in the credential profile; see the *Additional credential profile options* section in the ***Administration Guide***.

## 3.6 Configuring the Self-Service Request Portal

See the ***Derived Credentials Self-Service Request Portal*** guide for details of configuring your system to use in Self-Service Request Portal. In particular, refer to the *External identity providers* section for information on configuring the Self-Service Request Portal to use Microsoft Entra as an identity provider.

You are recommended to make the following specific changes to the `myid.json` configuration file:

- Map the Entra ID attribute `oid` to `XuSYSExternalReferenceId1` to ensure there is a mapping between the Entra object and the MyID account.

  **Note:** If you use a different field than `XuSYSExternalReferenceId1`, you must make sure that you configure the **External Entra Reference** option in the **External Systems** workflow to use the appropriate field; see section *3.7.3*, *Setting up the external system*.

- Map the Entra ID attribute `preferred_username` to `LogonName`; otherwise, the logon name is a numeric value, created at issuance.
- Set the `Mappings` node to include a roles mapping that includes the `Password User` role.

### 3.6.1 Example SSRP configuration file

The following example `myid.json` file contains example mappings for your SSRP system.

Placeholders are enclosed in square brackets `[]`.

See the *Configuring the Self-Service Request Portal for external identity providers* section in the ***Derived Credentials Self-Service Request Portal*** guide for more detailed information.

```
{
  "Providers":[
    {
      "Name":"Entra",
      "DisplayName":"Sign in with Microsoft",
      "Url":"https://login.microsoftonline.com/[tenant]/v2.0",
      "Enabled":true,
      "Type":"oid",
      "Default":true,
      "Scopes":"openid email profile",
      "ClientId":"[clientID]",
      "ClientSecretClear":"[clientsecret]",
      "ClientSecret":"[clientsecretencrypted]",
      "RetrieveUserInfo":true,
      "Mappings":[
        {
          "Match":{

          },
          "Attributes":[
            {
              "From":"oid",
              "To":"XuSYSExternalReferenceId1",
              "Mandatory":true,
              "Unique":true,
              "Update":true
            },
            {
              "From":"preferred_username",
              "To":"Email",
              "LookupExisting":true,
              "LdapSync":true
            },
            {
              "From":"preferred_username",
              "To":"LogonName"
            },
            {
              "From":"name",
              "To":"FullName"
            },
            {
              "From":"family_name",
              "To":"Surname",
              "Update":true
            },
            {
              "From":"given_name",
              "To":"FirstName"
            },
```

```
            {
              "From":"",
              "To":"Roles",
              "Static":"<role name='Derived Credential Owner' scope='1'/><role
name='Cardholder' scope='1'/><role name='PasswordUser' scope='1'/>"
            },
            {
              "From":"",
              "To":"GroupName",
              "Static":"Imported From Microsoft"
            },
            {
              "From":"",
              "To":"ParentGroupName",
              "Static":"External"
            }
          ]
        }
      ]
    }
  ]
}
```

## 3.7 Configuring MyID to issue passkeys using Microsoft Entra

To configure and enable this feature in MyID, you must:

- Set up the allowed origins in the MyID web.oauth2 settings.
- Set up an external system to communicate with the FIDO system.
- Create a credential profile for your FIDO devices.

### 3.7.1 Setting up the allowed origins

To allow MyID to perform attestation checks on a newly created FIDO credential with an origin that is not the MyID server, you must configure the MyID web.oauth2 settings to list the allowed origins for FIDO devices; this is the list of domains that are allowed to authenticate. You must include your MyID web server domain and the Microsoft domain.

To configure the server settings for multiple origins:

1. On the MyID web server, open the `appsettings.Production.json` file in a text editor.

   By default, this is:

   `C:\Program Files\Intercede\MyID\web.oauth2\appsettings.Production.json`

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

2. Edit the file to include the following:

   ```
   {
       "Fido":{
           "Config":{
               "Origins":["https://<server>", "https://login.microsoft.com"]
           }
       }
   }
   ```

   where:

   - `<server>` – the name of the MyID web server.

   **Important:** The `Origins` option is case sensitive, and must be consistent with the casing of the DNS Name in the web server's TLS certificate.

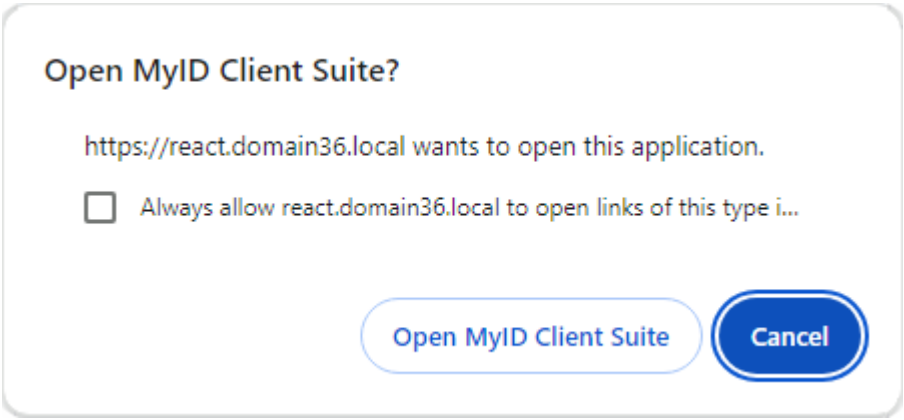
3. Save the `appsettings.Production.json` file.
4. Recycle the web service app pool:
   a. On the MyID web server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **myid.web.oauth2.pool** application pool, then from the pop-up menu click **Recycle**.

   This ensures that the web service has picked up the changes to the configuration file.

**Note:** If `Origins` is specified, it overrides any value in `Origin`.

### 3.7.2 Setting the MyID Client Service timeout

When you attempt to collect an externally-linked FIDO credential, if you have not already opened the MyID Client Service, the browser displays a pop-up asking you to open the MyID Client Suite (which contains the MyID Client Service required to collect the FIDO credential).




The browser waits ten seconds (by default) for you to click the option.

If the MyID Client Service does not run by the end of the timeout period, the browser displays the following error:

- OA10087 – The MyID client service encountered an issue while attempting to register your device with an external source.

If you want to change this timeout, you can configure the web.oauth2 settings file.

**Note:** This setting affects the timeout only when collecting externally-linked FIDO credentials; it does not affect the behavior of the pop-up in other circumstances.

To change the MyID Client Service timeout:

1. On the MyID web server, open the `appsettings.Production.json` file in a text editor.

   By default, this is:

   `C:\Program Files\Intercede\MyID\web.oauth2\appsettings.Production.json`

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

2. Edit the `MyID` section to include the following:

   ```
   {
       "MyID":  {
           "MaxCheckAliveTimeMs":  <timeout>
   }
   ```

   where:

   - `<timeout>` – the number of milliseconds to wait for the user to open the MyID Client Suite.

3. Save the `appsettings.Production.json` file.

4. Recycle the web service app pool:
   a. On the MyID web server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **myid.web.oauth2.pool** application pool, then from the pop-up menu click **Recycle**.

   This ensures that the web service has picked up the changes to the configuration file.

### 3.7.3 Setting up the external system

You must set up an external system to allow MyID to communicate with the FIDO system.

To configure a new FIDO Entra external system:

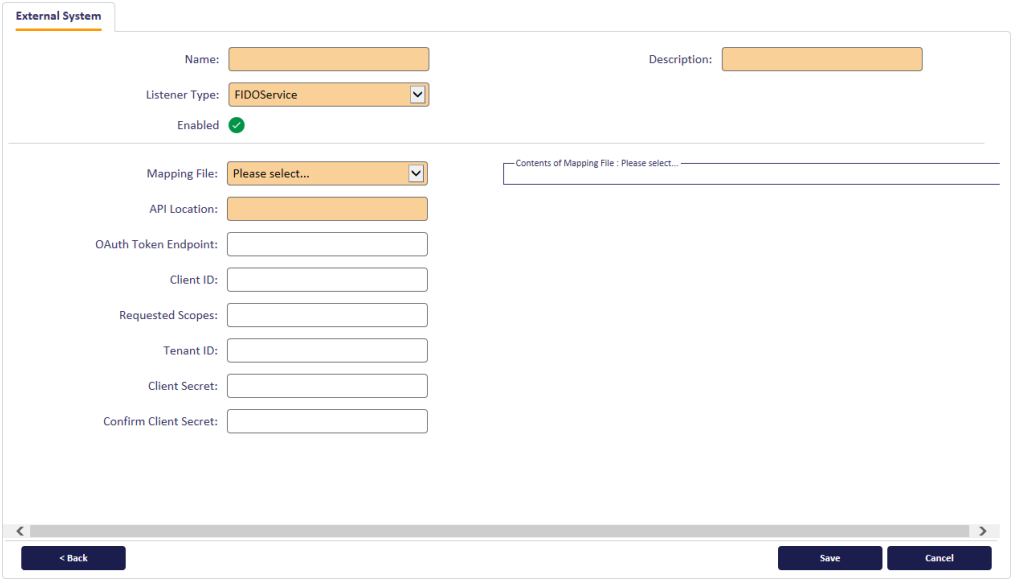
1. In MyID Desktop, from the **Configuration** category, select **External Systems**.

   You can also launch this workflow from the **Connections and Notifications** section of the **More** category in the MyID Operator Client. See the *Using Connections and Notifications workflows* section in the ***MyID Operator Client*** guide for details.

2. Click **New**.

3. From the **Listener Type** drop-down list, select **FIDOService**.

   The details for a FIDO external system appear.




4. Complete the following details:

   - **Name** – Type the name of the external system.

     Take a note of this name; you use the external system name to identify the appropriate external system when you configure the credential profile for FIDO devices.

   - **Description** – Type a description for the external system.

   - **Enabled** – Select this option to enable the external system, or deselect it to disable the external system. When the external system is disabled, MyID does not attempt to communicate with the external system.

   - **Mapping File** – Select **RESTFidoEntra**.

     The contents of the `RESTFidoEntra.xml` file are displayed. When you select this file, an additional option appears on screen: **External Entra Reference**.

- **API Location** – Type the URL of the FIDO service API.

  For the beta version of the Entra passkey service, use:

  `https://graph.microsoft.com/beta`

  Once the feature has been released, change this to:

  `https://graph.microsoft.com/v1.0`

  You can check the status of the API at the Microsoft website:

  ***learn.microsoft.com/en-us/graph/api/resources/fido2authenticationmethod?view=graph-rest-beta***
- **OAuth Token Endpoint** – Type the location of the OAuth token endpoint.

  You can include the Entra tenant ID in this URL, or you can include a placeholder that MyID substitutes for the value in the **Tenant ID** field; for example:

  `https://login.microsoftonline.com/{TenantID}/oauth2/v2.0/token`
- **Client ID** – Type the client identifier for the FIDO service.
- **Requested Scopes** – Type the scopes that must be granted to the bearer token for the FIDO service.
- **Tenant ID** – Type the ID of the tenant to be used for the FIDO service. This value is substituted for `{TenantID}` in the **OAuth Token Endpoint**.
- **External Entra Reference** – select the field in the `vPeopleUserAccounts` view in the MyID database in which you want to store the Entra `ObjectGUID` for the user. By default, MyID uses the `XuSYSExternalReferenceId1` field. Select one of the following options:
  - **XuSYSExternalReferenceId1**
  - **XuSYSExternalReferenceId2**
  - **XuSYSExternalReferenceId3**
  - **UserPrincipalName**

  If you select an option other than `XuSYSExternalReferenceId1`, you must make sure that you have stored the Entra ID in the appropriate field (for example, using the Self-Service Request Portal, the MyID Operator Client, the MyID Core API, or directly in the database).
- **Client Secret** – Type the client secret to use when authenticating to the FIDO service.
- **Confirm Client Secret** – Confirm the client secret to use when authenticating to the FIDO service.

5. Click **Save**.

### 3.7.4 Creating a credential profile for passkeys

Once you have configured your external system, you can create a credential profile for passkeys that use this external system. See section *5.2*, *Setting up a passkey credential profile for the Self-Service Request Portal*.

# 4 Enterprise attestation

Attestation is the process of obtaining proof of authenticity from a passkey device; by default, the basic attestation process allows an authenticator to provide only its AAGUID (a 128-bit identifier of the type of the authenticator) and high-level information about its type and capabilities.

Enterprise attestation extends this process by including a unique identifier for the authenticator, such as its serial number. This allows your organization to control the issuance of passkeys to approved authenticators.

**Important:** You must discuss your enterprise attestation requirements with your authenticator vendor so they can supply you with devices that are suitable for your purposes. You may require devices that have been manufactured to your organization's requirements; your vendor can advise you.

The following types of enterprise attestation are available:

- Vendor-facilitated enterprise attestation.

  If you want to use vendor-facilitated enterprise attestation, you must work with an authenticator vendor to produce devices with a specific set of keys to include enterprise attestation metadata. You can then configure MyID CMS to issue passkeys to only those devices that pass the enterprise attestation checks, ensuring that you are issuing passkeys only to the approved devices.

  A major component of vendor-facilitated enterprise attestation is a preconfigured list of relying party (RP) IDs that is built into the devices by the manufacturer; that is, a list of the domains that are allowed to request enterprise attestation from a device.

  Vendor-facilitated enterprise attestation is supported using most modern browsers (for example, Chrome, Firefox, and Edge) and the MyID Client Service app.

- Platform-managed enterprise attestation

  As an alternative to manufacturing devices with a preconfigured list of RP IDs, you can instead manage and control the list of relying parties using an enterprise-managed browser.

  Currently, this is supported only using Google Chrome (as an experimental feature) and the MyID Client Service app.

  For information on configuring your organization's Chrome browsers to provide the allowed list of RP IDs, see section *4.1*, *Enabling platform-managed enterprise attestation in Google Chrome*.

To configure MyID to issue passkeys using enterprise attestation, you must configure a credential profile with the **Require Attestation** option set to **Enterprise** or **Enterprise (Restricted)**. See section *5.1*, *Setting up a passkey credential profile for the MyID Operator Client* or section *5.2*, *Setting up a passkey credential profile for the Self-Service Request Portal* for details.

Depending on the implementation of enterprise attestation by your passkey device manufacturer, the device serial number may be extracted from the device to provide unique identification; see section *4.2*, *Linked credentials* for details.

When you issue passkeys using enterprise attestation, this affects how you can work with the passkeys, in particular when canceling a credential or marking the device as lost or disposed. See section *4.3*, *Working with enterprise attestation credentials*.

## 4.1 Enabling platform-managed enterprise attestation in Google Chrome

A scenario where you might want to use platform-managed enterprise attestation is if your devices have been manufactured with an attestation certificate that is available in the online global MDS3 FIDO Alliance metadata; you can then configure your organization's enterprise-managed browsers to provide a list of allowed relying party IDs rather than requiring the list of allowed domains to be encoded on the device.

Platform-managed enterprise attestation is supported using the Google Chrome browser and the MyID Client Service app.

**Note:** Google Chrome is the only browser that currently supports this feature.

To enable this feature on Chrome:

1. Open Chrome, and type the following in the location bar:

   `chrome://flags/#web-authentication-permit-enterprise-attestation`

2. In the **Web Authentication Enterprise Attestation** section:

   a. Add the MyID server domain to the comma-delimited list in the text box.

   This must be the full domain name including the `https` prefix; for example:

   `https://myserver.example.com`

   b. Select **Enabled** from the drop-down list.

## 4.2 Linked credentials

When you use enterprise attestation, MyID CMS may be able to extract a serial number from the authenticator that uniquely identifies that physical device. This depends on the capabilities of the device and how it was manufactured.

This serial number may also be the same as the serial number obtained during PIV credential issuance, again depending on the manufacturer.

If this serial number is available, MyID CMS can use it to identify all issued credentials that are issued to the same physical device; for example, you may have multiple passkeys on the same device, or a PIV credential and a passkey.

When MyID CMS records the serial number for a passkey, it uses the base device serial number and appends a number based on the number of credentials issued to the device.

To view linked credentials:

1. Search for a device, and view its details.

   You can search for all credentials that share the same physical device by using the physical device's **Serial Number** with a wildcard; for example:

   ```
   12345678*
   ```

   returns all credentials issued to the physical device with serial number `12345678`.

   Alternatively, insert the device into a reader.

   You can also view a device from any form that contains a link to the device.

   For example:

   - Click the item in the list on the **Devices** tab of the View Person form.
   - Click the link icon on the **Device Serial Number** field of the View Request form.

2. Click the **Linked Credentials** tab.

   This tab lists all of the credentials on the same device that MyID CMS can correlate using their serial number.

   You can click on a device in the list to open its View Device page.

## 4.3 Working with enterprise attestation credentials

Issuing your passkeys using enterprise attestation affects the lifecycle operations you can carry out on the device.

### 4.3.1 Issuing passkeys

When you attempt to register a passkey using credential profile configured for enterprise attestation, if MyID CMS does not trust the enterprise attestation certificate, it rejects the issuance of the passkey.

### 4.3.2 Authenticating passkeys

When you attempt to authenticate a passkey that has been issued for enterprise attestation, if the relying party does not trust the enterprise attestation certificate, it rejects the authentication.

**Note:** You must check that your relying party supports enterprise attestation.

If you are using platform-managed enterprise attestation, in addition to trusting the enterprise attestation certificate, the relying party must also be listed in the platform-managed RP ID list.

### 4.3.3 Resetting and erasing devices

You may find that the enterprise attestation feature on your devices is disabled if you reset the device; see the instructions provided by your device manufacturer for details of re-enabling this feature if required. For example, you can use an app provided by your device manufacturer such as the Yubico Authenticator app.

**Note:** For some devices (for example, YubiKey v57), MyID CMS carries out a reset on the device when erasing it. Therefore, this operation may also disable the enterprise attestation feature, requiring you to re-enable the feature before you can use the device for enterprise attestation again.

### 4.3.4 Lost or disposed authenticators

If you have an authenticator that is enabled for enterprise attestation, and you set the disposal status of a passkey on the device to **Lost** or **Disposed**, you can no longer issue any passkeys to that physical device.

If you want to issue passkeys to that device, you must reset the disposal status; for example, if a lost device is subsequently found.

**Note:** When you set the disposal status for a passkey, it does not affect the other passkeys on the device. To set the disposal status for all passkeys on a device, you can use the batch feature; see the *Setting the disposal status of multiple devices* section in the ***MyID Operator Client*** guide.

### 4.3.5 Canceling credentials

When you cancel a passkey, it does not affect the other passkeys on the device. To cancel all passkeys on a device, you can use the batch feature; see the *Canceling multiple devices* section in the ***MyID Operator Client*** guide.

# 5 Setting up credential profiles for passkeys

You must set up one or more credential profiles for your passkeys. The options you select depend on whether you intend to request the passkey through the MyID Operator Client or through the Self-Service Request Portal.

This chapter contains instructions for setting up your credential profiles.

You can:

- Create a credential profile for passkeys that you can request and collect through the MyID Operator Client.

  This method allows you to request and collect passkeys for users who may or may not already have credentials issued to them.

  See section *5.1*, *Setting up a passkey credential profile for the MyID Operator Client*.
- Create a credential profile for passkeys that you can request and collect through the Self-Service Request Portal.

  This method allows you to use an existing credential as the basis for your passkey, or to use authentication with Microsoft Entra as the basis for your passkey.

  See section *5.2*, *Setting up a passkey credential profile for the Self-Service Request Portal*.

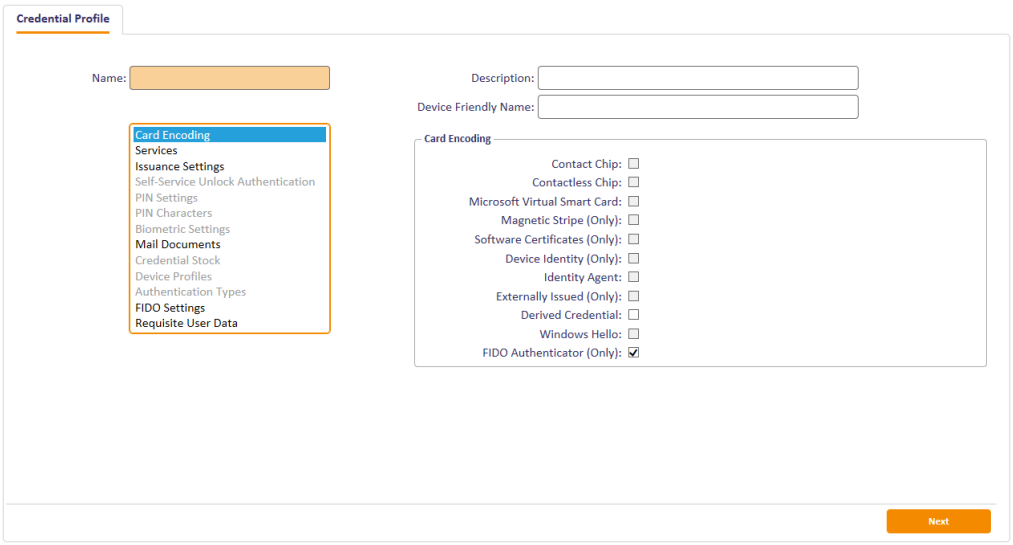
## 5.1 Setting up a passkey credential profile for the MyID Operator Client

To set up a credential profile for passkeys that you can use for requests made in the MyID Operator Client:

1. Log on to MyID Desktop as an administrator.
2. From the **Configuration** category, select **Credential Profiles**.
3. Click **New**.

4. In the **Card Encoding** list, select **FIDO Authenticator (Only)**.

   **Note:** Most other options are disabled. The **Derived Credential** option is not disabled; however, it is used only for requests made through the Self-Service Request Portal. See section *5.2*, *Setting up a passkey credential profile for the Self-Service Request Portal*.




5. In the **Services** section, you can set the following:

   - **MyID Logon** – select this option if you want to be able to log on to MyID with the authenticator.

   **Note:** The **MyID Encryption** option is disabled. You cannot use a passkey to store an encryption certificate.

6. In the **Issuance Settings** section, the following options are available:
   - **Validate Issuance**
   - **Validate Cancellation** – do not select this option. Validating cancellation is not supported with passkeys, and setting this option may result in being unable to cancel the device.
   - **Lifetime**
   - **Credential Group**
   - **Block Multiple Requests for Credential Group**
   - **Cancel Previously Issued Device**
   - **Enforce Photo at Issuance** – do not select this option. Request checks are performed for passkeys, but issuance checks are not; instead of standard MyID issuance, authenticators use a FIDO-specific registration process.
   - **Notification Scheme**
   - **Require user data to be approved**

   See the *Working with credential profiles* section in the ***Administration Guide*** for details of these options.

   You must also set the following option:

   - **Generate Code on Request** – set this to one of the following options:
     - **Simple Logon Code** – the FIDO registration code is generated using the complexity rules as defined by the **Simple Logon Code Complexity** configuration option on the **Auth Code** tab of the **Security Settings** workflow.

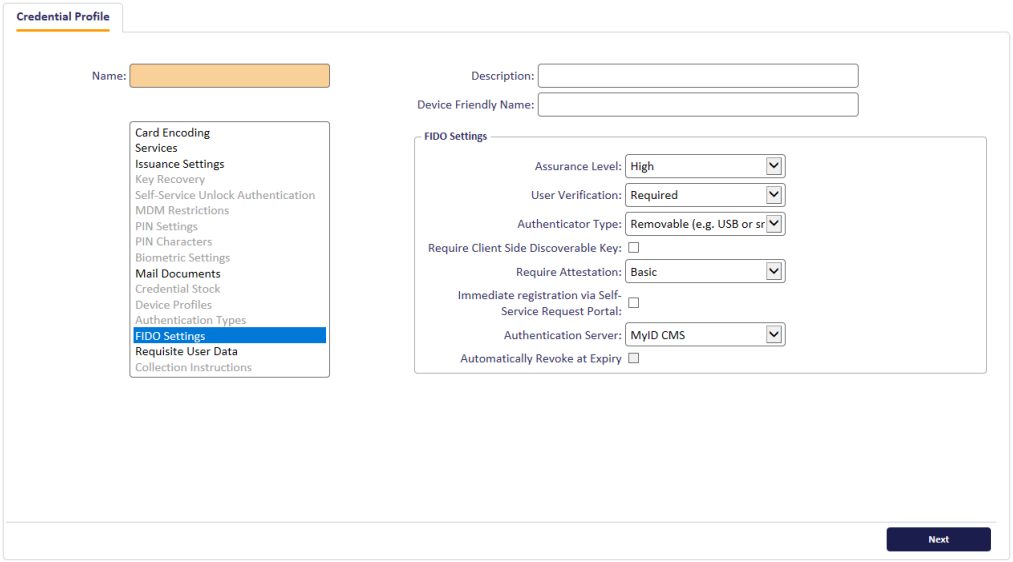
       By default, this is `12-12N`, which means a 12-digit number.
     - **Complex Logon Code** – the FIDO registration code is generated using the complexity rules as defined by the **Complex Logon Code Complexity** configuration option on the **Auth Code** tab of the **Security Settings** workflow.

       By default, this is `12-12ULSN[BGIlOQDSZ]`, which means a 12-character code containing upper case, lower case, special characters, and numbers, and a set of commonly-confused characters excluded.

   **Important:** Do not select **None**. MyID must generate a FIDO registration code to be used in the passkey registration process.

   For more information about the format of these codes, see the *Setting up logon codes* section in the ***Administration Guide***.

7. In the **FIDO Settings** section, set the following:




- **Assurance Level** – select one of the following options:
  - **Basic** – the passkey uses single factor authentication, and is suitable for use with some external systems, but not for access to crucial systems.
  - **High** – the passkey uses multi-factor authentication, and is suitable for use with secure systems, such as logging on to MyID.

    You are recommended to set **Assurance Level** to **High** only when you have also set the **User Verification** to **Required**.

  MyID differentiates between passkeys that have been issued with a credential profile where the **Assurance Level** is set to **Basic** or **High** – for example, you can enable logon to MyID for **FIDO High Assurance**, but disable logon for **FIDO Basic Assurance**. See section *2.6*, *Configuring MyID for logon with passkeys* for details.
- **User Verification** – select one of the following options:
  - **Required** – the passkey supports two-factor authentication. If the authenticator does not support two-factor authentication, it cannot be registered.
  - **Preferred** – the passkey will use two-factor authentication if the authenticator supports that feature, but will still be registered if it supports only one-factor authentication.
  - **Discouraged** – the passkey will use single-factor authentication, unless the authenticator cannot work without multi-factor authentication.
- **Authenticator Type** – select one of the following options:
  - **Internal** – you can issue this credential profile to internal passkeys; for example, authenticators included in mobile devices such as cell phones.
  - **Removable** – you can issue this credential profile to external removable authenticators; for example, USB tokens or smart cards.
  - **Internal or Removable** – you can issue this credential profile to internal or removable passkeys.

- **Require Client Side Discoverable Key** – select this option to ensure that the passkey supports Resident Keys. If you select this option, and the passkey supports client side discoverable keys, you can choose not to provide the username manually when using the passkey to log on to MyID; see section *6.5*, *Signing in to MyID CMS with a passkey*.
- **Require Attestation** – select the level of attestation check to carry out during the registration process:
  - **None** – do not carry out any attestation checks.
  - **Basic** – carry out an attestation check during the registration process.
  - **Basic (Restricted)** – carry out an attestation check during the registration process, using only a local metadata repository (either `MDSCacheDirPath` or `MDSCacheDirPathEnterprise`).

    See section *2.3.1*, *Setting up a local metadata repository* for details.
  - **Enterprise** – carry out an enterprise attestation check during the registration process.

    MyID attempts to extract an enterprise attestation serial number from the attestation certificate. If it cannot obtain a serial number, it uses the data from a local metadata repository that is configured using the `MDSCacheDirPathEnterprise` path for the attestation check.

    See section *4*, *Enterprise attestation* for details of enterprise attestation.
  - **Enterprise (Restricted)** – carry out an enterprise attestation check during the registration process, using only the data from a local metadata repository that is configured using the `MDSCacheDirPathEnterprise` path for the attestation check.

  **Note:** In previous releases, this was a single option labeled **Enforce Authenticator Attestation Check**. If you upgrade from a system that used this option, any credential profiles that did not have this option selected are set to **None**, and those that did have this option selected are set to **Basic**.
- **Immediate registration via Self-Service Request Portal** – used only for requests made through the Self-Service Request Portal. See section *5.2*, *Setting up a passkey credential profile for the Self-Service Request Portal*.
- **Authentication Server** – select the authentication server for your passkeys.

  By default, this is set to MyID CMS, which means that you use MyID as the authentication server for your FIDO devices.

  If you are using an external authentication server for your passkeys (for example, Entra), select the name of the external system you created from the drop-down list.
- **Automatically Revoke at Expiry** – if you have selected an **Authentication Server** other than MyID CMS, you can specify that you want the passkey to be revoked automatically when it expires. At the credential expiry time, MyID cancels the credential in both MyID and on the external authentication server.

  **Note:** The expiry cancellation job runs every 30 minutes, so there may be a delay between the expiry time and the actual revocation.

8. In the **Requisite User Data** section, set any user attributes that you want to require for the people who will request passkeys.

   For example, as the FIDO notification is sent as an email, you are recommended to select **Email** in the **Required for Request** column.

   If you have configured your system to send the registration code in an SMS, you are recommended to select **Cell** in the **Required for Request** column.

   For more information about this features, see the *Requisite User Data* section in the ***Administration Guide***.

9. Click **Next**.

10. In the **Select Roles** screen, select the roles you want to be able to receive, request, or validate FIDO registrations.

    - Make sure that people who will receive the passkey have a role that is selected in the **Can Receive** list.
    - Make sure that operators who will request passkeys have a role that is selected in the **Can Request** list.
    - If you have selected the **Validate Issuance** option, make sure that operators who will approve requests for passkeys have a role that is selected in the **Can Validate** list.

    **Note:** You do not need to select any roles in the **Can Collect** list. Collecting passkeys is carried out by the person who is receiving the authenticator using a self-service registration process.

11. Click **Next**.

12. Type your **Comments**, then click **Next** to save the credential profile and complete the workflow.

## 5.2 Setting up a passkey credential profile for the Self-Service Request Portal

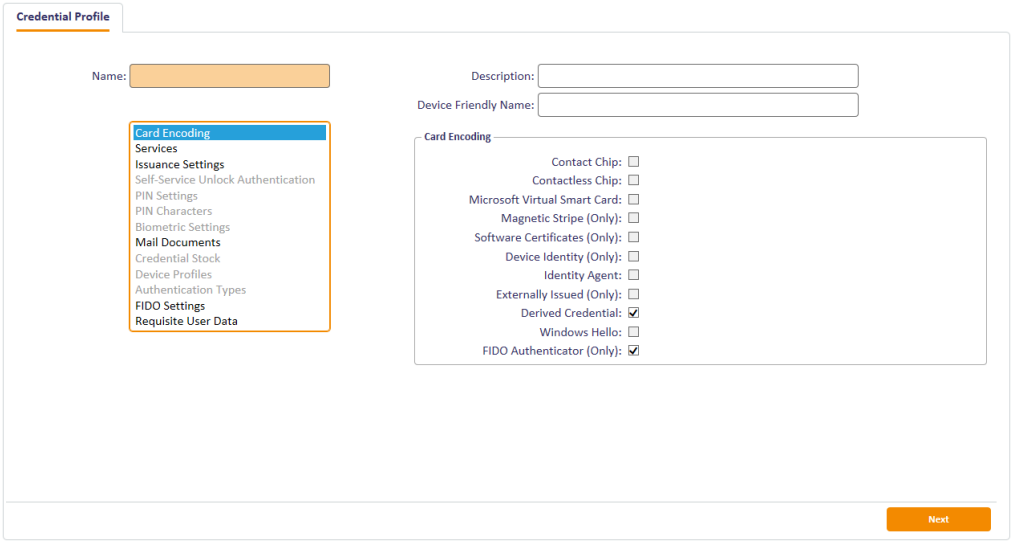
To set up a credential profile for FIDO authenticators that you can use for requests made in the Self-Service Request Portal:

1. Log on to MyID Desktop as an administrator.
2. From the **Configuration** category, select **Credential Profiles**.
3. Click **New**.
4. Type a **Name** and **Description** for the credential profile.
5. Optionally, type a **Device Friendly Name**.

   For Entra passkeys, If the credential profile has a device friendly name, it is used as the device name in Entra, truncated to 30 characters (which is a limitation of Entra). If the credential profile does not have a device friendly name, the name in Entra is set to `MyID Passkey`.

6. In the **Card Encoding** list, select the following:
   - **Derived Credential**
   - **FIDO Authenticator (Only)**

**Note:** The other options are disabled.




7. In the **Services** section, you can set the following:

   - **MyID Logon** – select this option if you want to be able to log on to MyID with the authenticator.

   **Note:** The **MyID Encryption** option is disabled. You cannot use a FIDO Authenticator to store an encryption certificate.

8. In the **Issuance Settings** section, the following options are available:
   - **Validate Issuance**
   - **Validate Cancellation** – do not select this option. Validating cancellation is not supported with FIDO authenticators, and setting this option may result in being unable to cancel the device.
   - **Lifetime**
   - **Credential Group**
   - **Block Multiple Requests for Credential Group**
   - **Cancel Previously Issued Device**
   - **Enforce Photo at Issuance** – do not select this option. Request checks are performed for FIDO authenticators, but issuance checks are not; instead of standard MyID issuance, authenticators use a FIDO-specific registration process.
   - **Notification Scheme**
   - **Require user data to be approved**

   See the *Working with credential profiles* section in the ***Administration Guide*** for details of these options.

   You must also set the following option:

   - **Generate Code on Request** – set this to one of the following options:
     - **Simple Logon Code** – the FIDO registration code is generated using the complexity rules as defined by the **Simple Logon Code Complexity** configuration option on the **Logon** tab of the **Security Settings** workflow.

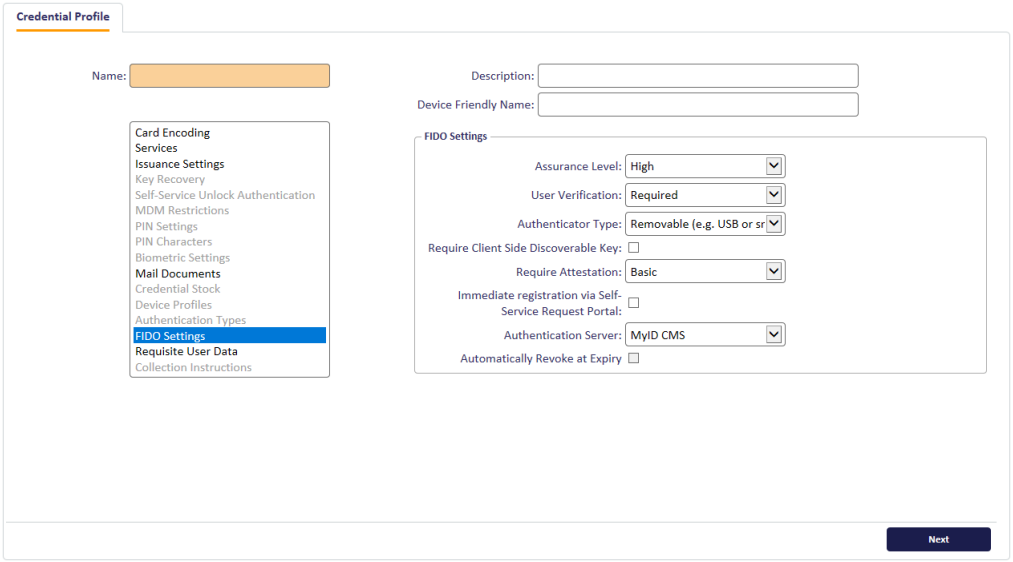
       By default, this is `12-12N`, which means a 12-digit number.
     - **Complex Logon Code** – the FIDO registration code is generated using the complexity rules as defined by the **Complex Logon Code Complexity** configuration option on the **Auth Code** tab of the **Security Settings** workflow.

       By default, this is `12-12ULSN[BGIlOQDSZ]`, which means a 12-character code containing upper case, lower case, special characters, and numbers, and a set of commonly-confused characters excluded.

   **Important:** Do not select **None**. MyID must generate a FIDO registration code to be used in the FIDO authenticator registration process.

   For more information about the format of these codes, see the *Setting up logon codes* section in the ***Administration Guide***.

9. In the **FIDO Settings** section, set the following:




- **Assurance Level** – select one of the following options:
  - **Basic** – the FIDO authenticator uses single factor authentication, and is suitable for use with some external systems, but not for access to crucial systems.
  - **High** – the FIDO authenticator uses multi-factor authentication, and is suitable for use with secure systems, such as logging on to MyID.

    You are recommended to set **Assurance Level** to **High** only when you have also set the **User Verification** to **Required**.

  MyID differentiates between FIDO authenticators that have been issued with a credential profile where the **Assurance Level** is set to **Basic** or **High** – for example, you can enable logon to MyID for **FIDO High Assurance**, but disable logon for **FIDO Basic Assurance**. See section *2.6*, *Configuring MyID for logon with passkeys* for details.
- **User Verification** – select one of the following options:
  - **Required** – the FIDO authenticator supports two-factor authentication. If the authenticator does not support two-factor authentication, it cannot be registered.
  - **Preferred** – the FIDO authenticator will use two-factor authentication if the authenticator supports that feature, but will still be registered if it supports only one-factor authentication.
  - **Discouraged** – the FIDO authenticator will use single-factor authentication, unless the authenticator cannot work without multi-factor authentication.
- **Authenticator Type** – select one of the following options:
  - **Internal** – you can issue this credential profile to internal FIDO authenticators; for example, authenticators included in mobile devices such as cell phones.
  - **Removable** – you can issue this credential profile to external removable authenticators; for example, USB tokens or smart cards.
  - **Internal or Removable** – you can issue this credential profile to internal or removable FIDO authenticators.

- **Require Client Side Discoverable Key** – select this option to ensure that the FIDO authenticator supports Resident Keys. If you select this option, and the FIDO authenticator supports client side discoverable keys, you can choose not to provide the username manually when using the FIDO authenticator to log on to MyID; see section *6.5*, *Signing in to MyID CMS with a passkey*.
- **Require Attestation** – select the level of attestation check to carry out during the registration process:
  - **None** – do not carry out any attestation checks.
  - **Basic** – carry out an attestation check during the registration process.
  - **Basic (Restricted)** – carry out an attestation check during the registration process, using only a local metadata repository (either `MDSCacheDirPath` or `MDSCacheDirPathEnterprise`).

    See section *2.3.1*, *Setting up a local metadata repository* for details.
  - **Enterprise** – carry out an enterprise attestation check during the registration process.

    See section *4*, *Enterprise attestation* for details of enterprise attestation.
  - **Enterprise (Restricted)** – carry out an enterprise attestation check during the registration process, using only a local metadata repository that is configured using the `MDSCacheDirPathEnterprise` path.

  **Note:** In previous releases, this was a single option labeled **Enforce Authenticator Attestation Check**. If you upgrade from a system that used this option, any credential profiles that did not have this option selected are set to **None**, and those that did have this option selected are set to **Basic**.
- **Immediate registration via Self-Service Request Portal** – select this option if you want to register the authenticator immediately when the cardholder makes the request in the Self-Service Request Portal. If you do not select this option, MyID sends the standard registration messages, and the person can register their authenticator later.

  **Important:** If you are using Entra for authentication for your passkeys, you must select this option; If you do not set this option, instead of registering the device immediately, SSRP displays a message that a request has been created and sends an email notification.
- **Authentication Server** – select the authentication server for your passkeys.

  By default, this is set to MyID CMS, which means that you use MyID as the authentication server for your FIDO devices.

  If you are using an external authentication server for your passkeys (for example, Entra), select the name of the external system you created from the drop-down list.
- **Automatically Revoke at Expiry** – if you have selected an **Authentication Server** other than MyID CMS, you can specify that you want the passkey to be revoked automatically when it expires. At the credential expiry time, MyID cancels the credential in both MyID and on the external authentication server.

  **Note:** The expiry cancellation job runs every 30 minutes, so there may be a delay between the expiry time and the actual revocation.

10. In the **Requisite User Data** section, set any user attributes that you want to require for the people who will request FIDO authenticators.

    For example, if you are not using immediate registration, as the FIDO notification is sent as an email, you are recommended to select **Email** in the **Required for Request** column.

    If you have configured your system to send the registration code in an SMS, you are recommended to select **Cell** in the **Required for Request** column.

    For more information about this features, see the *Requisite User Data* section in the ***Administration Guide***.

11. Click **Next**.

12. In the **Select Roles** screen, select the **Derived Credential Owner** role for each of the following:

    - **Can Receive**
    - **Can Request**
    - **Can Collect**

    **Note:** You do not need to select any of the roles held by the person who will receive the FIDO registration request.

13. Click **Next**.

14. Type your **Comments**, then click **Next** to save the credential profile and complete the workflow.

# 6 Working with passkeys

Once you have configured MyID for passkeys, you can:

- Request passkeys for people.

  See section *6.1*, *Requesting passkeys*.
- Register your passkey with MyID.

  See section *6.2*, *Registering passkeys*.
- View passkeys in MyID CMS or in Entra.

  See section *6.3*, *Viewing passkeys*.
- Enable, disable, or cancel passkeys.

  See section *6.4*, *Enabling, disabling, and canceling passkeys*.
- Log on to MyID using your registered passkey.

  See section *6.5*, *Signing in to MyID CMS with a passkey*.
- Sign in to Microsoft services using your Entra-registered passkey.

  See section *6.6*, *Signing in to Microsoft using a passkey*.

## 6.1 Requesting passkeys

You can request a passkey for a person using the MyID Operator Client. Alternatively, if a person already has a smart card issued, they can use the Self-Service Request Portal to request (and optionally register) a passkey for themselves.

**Note:** A primary requirement for MyID to issue passkeys with Microsoft Entra is that the `ObjectGUID` of the user's account in Entra must exist in the MyID CMS database. If you are using Entra for authentication, you must either add the Entra ID to the MyID CMS database, or use the Self-Service Request Portal configured for Entra external authentication; see section *3*, *Issuing passkeys with Microsoft Entra*.

### 6.1.1 Requesting passkeys using the MyID Operator Client

You can use the MyID Operator Client to request a passkey for a person.

1. Log on to the MyID Operator Client.
2. Click the **People** category and search for a person.

   See the *Searching for a person* section in the ***MyID Operator Client*** guide.
3. Click the **Request Device** option in the button bar at the bottom of the screen.

   You may have to click the **...** option to see any additional available actions.
4. From the **Credential Profile** drop-down list, select the FIDO credential profile you want to use.

   See section 5, *Setting up credential profiles for passkeys* for details of FIDO credential profiles.
5. Click **Save** to make the request.

For more information about requesting devices, see the *Requesting a device for a person* section in the ***MyID Operator Client*** guide.

If your FIDO credential profile is configured to require validation, you must approve the request before MyID notifies the person that they can register their passkey; see the *Approving requests* section in the ***MyID Operator Client*** guide for details.

See section 6.2, *Registering passkeys* for details of carrying out the registration process.

### 6.1.2 Requesting passkeys using the Self-Service Request Portal

If you have an already-issued smart card, you can use this to request a passkey through the Self-Service Portal.

For information on configuring the Self-Service Request Portal, see the ***Derived Credentials Self-Service Request Portal*** guide.

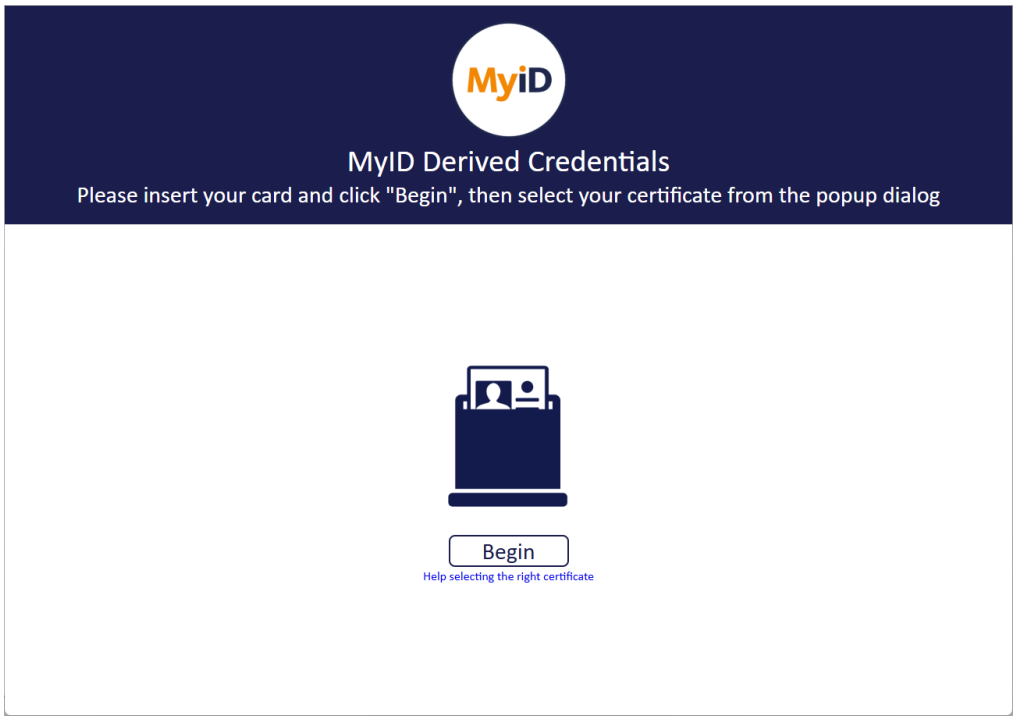
To request a passkey through the Self-Service Portal:

1. Open a web browser and navigate to the StartPage on the SSRP web server:

   `https://<myserver>/StartPage`

   where `<myserver>` is the address of the MyID server hosting the Self-Service Portal.

   The start page appears.




2. Insert your card, click Begin, then select a certificate from your card.

   The credential profile selection page appears.

3. Select the credential profile you want to use.

   See section *5*, *Setting up credential profiles for passkeys* for details of setting up your passkey credential profiles.

   The next stage depends on how you have set up the **Immediate registration via Self-Service Request Portal** option in the credential profile:

   - If the **Immediate registration via Self-Service Request Portal** option is set, you can register your passkey immediately; see section *6.2.2*, *Registering passkeys using the Self-Service Request Portal*.
   - If the **Immediate registration via Self-Service Request Portal** option is *not* set, MyID sends a registration link and a registration code; see section *6.2.1*, *Registering passkeys through notifications*.

## 6.1.3 Requesting passkeys using the Self-Service Request Portal using Entra for authentication

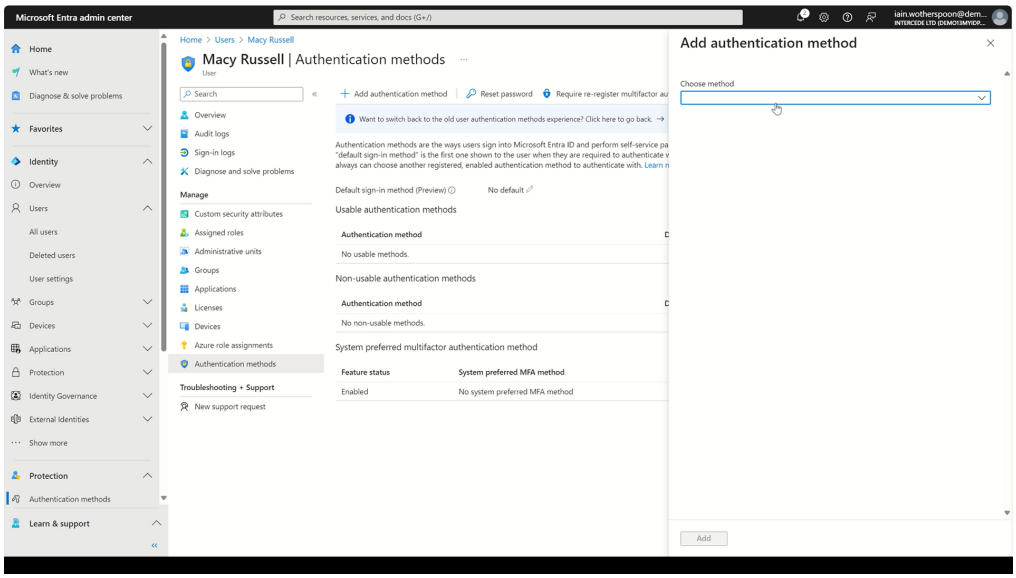
The recommended method of registering a passkey using Entra for authentication for this release is through the MyID Self-Service Request Portal (SSRP). This procedure imports your user account into MyID, or links it to your existing MyID account, then guides you through the process of registering your passkey.

You must be able to authenticate to Entra; for example, you can request a Temporary Access Pass (TAP) for Entra and then use Entra to authenticate to the SSRP and request your passkey.

### 6.1.3.1 Requesting a temporary access pass

To request a TAP:

1. In the Microsoft Entra admin center, select the user to whom you want to issue the TAP.
2. In the **Authentication methods** section, click **Add authentication method**.




3. From the **Choose method** drop-down list, select **Temporary Access Pass**.
4. Select the options for the TAP, including the duration, then click **Add**.

   The user now has a TAP they can use to authenticate to Entra.

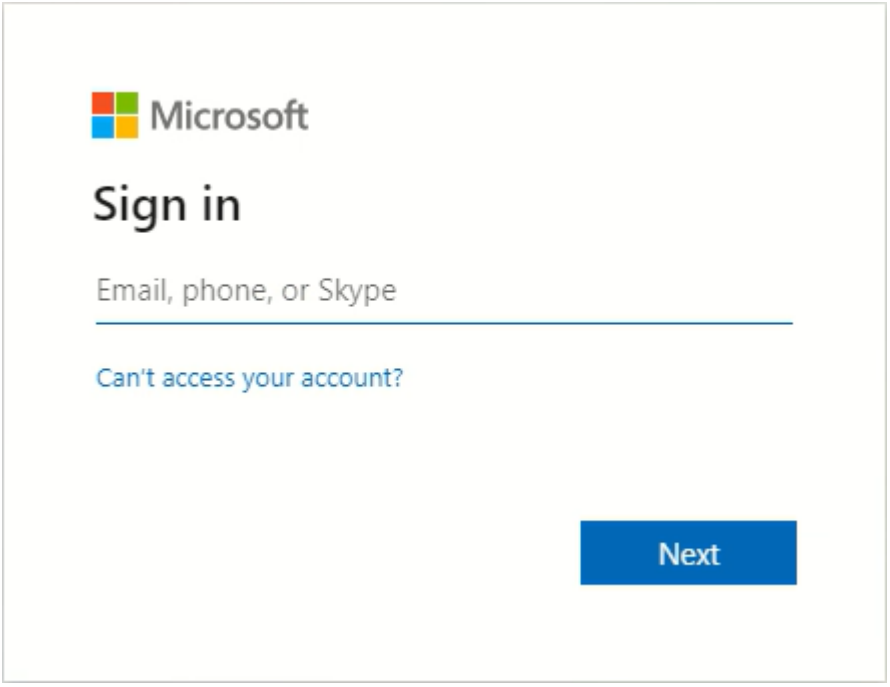
### 6.1.3.2 Requesting a passkey using your TAP

Once you have a TAP, you can request your passkey.

To request your passkey:

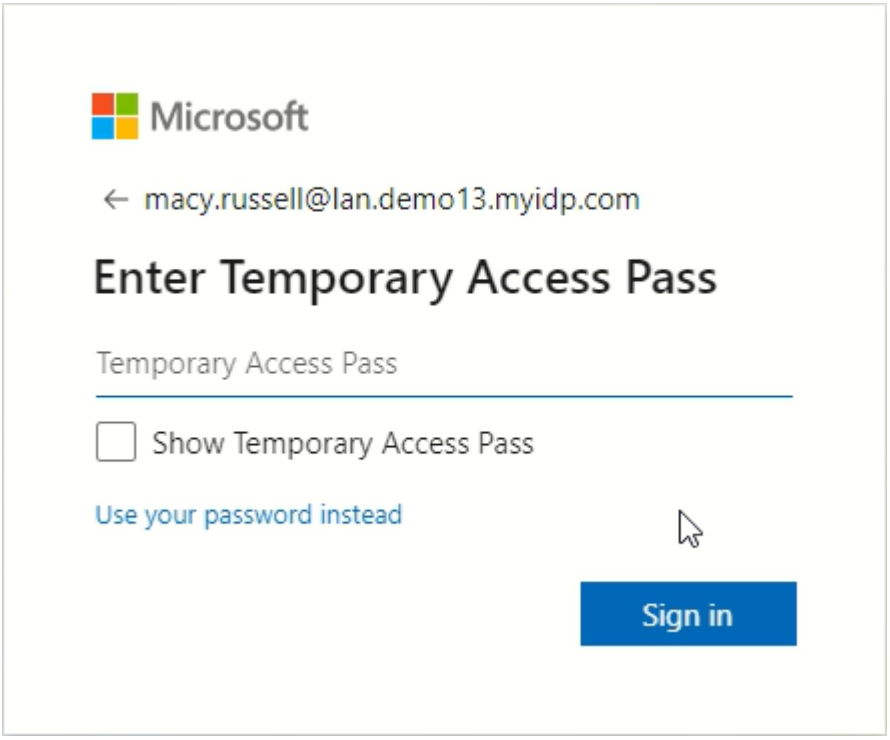
1. Click the provided link to the SSRP.

   Your administrator may provide you with a link that takes you directly to the Entra login, or you may have to select the identity provider manually.



2. On the Microsoft Sign In page, provide your email address, and click **Next**.

   The Enter Temporary Access Pass screen appears.

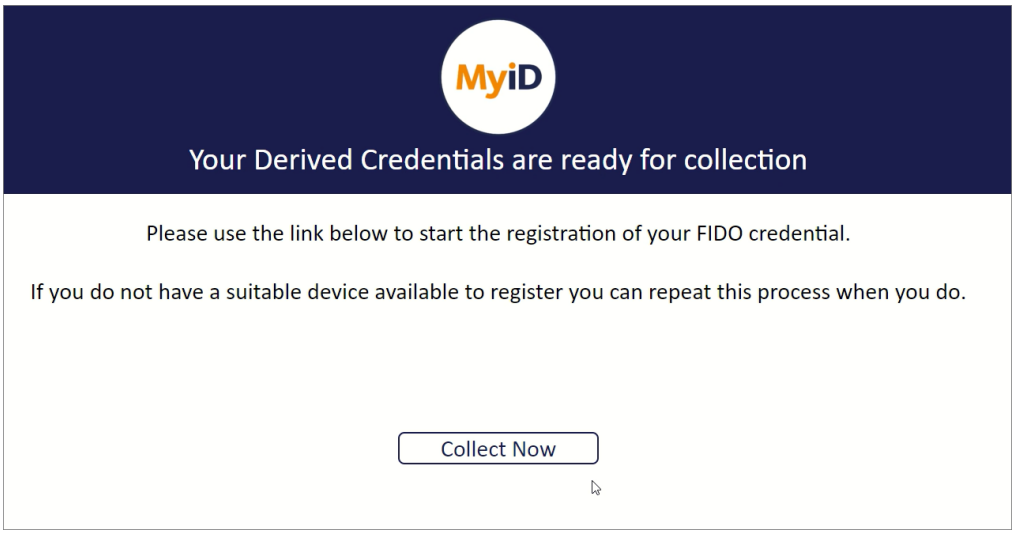



3. Type your TAP and click **Sign in**.

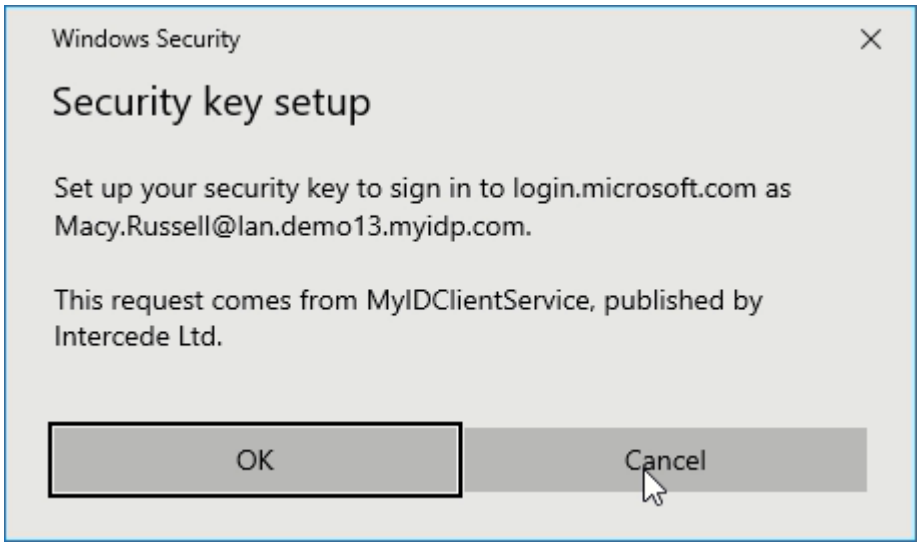
   You can choose whether to stay signed in; click **Yes** to stay signed in, or **No** to sign out after this operation.

   You are now returned to the SSRP web page, having authenticated using your Microsoft Entra account.

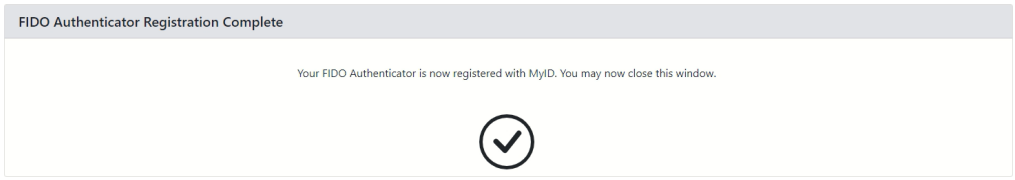
4. If you have more than one credential profile available, select the credential profile you want to use to issue your new passkey.



5. Click the **Collect Now** link to begin the collection process.




6. Follow the instructions provided by the Windows Security system to set up your device; for example, enter your new PIN and touch the device.




7. Once the process is complete, close the window.

   Your passkey is now ready for use.

## 6.2 Registering passkeys

You can register your passkey using the following methods:

- Using a link and a registration code provided through notifications to your email address or cell phone.
- Using the Self-Service Request Portal for immediate registration after requesting an authenticator.

### 6.2.1 Registering passkeys through notifications

MyID sends two notifications to the person when a passkey has been requested for them:

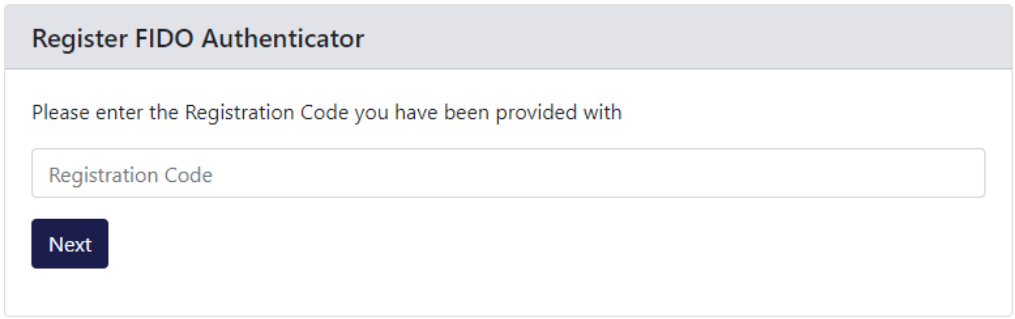
- An email message containing a link to the registration web page.
- An email message (or SMS, depending on configuration) containing a single-use registration code.

To register your passkey:

1. Click the link in the FIDO registration email.

   This should take you to a web page with an URL similar to:

   `https://<myserver>/web.oauth2/fido/register/begin?requestId=BADF1894-266B-4B80-9084-6ECD721347BD`



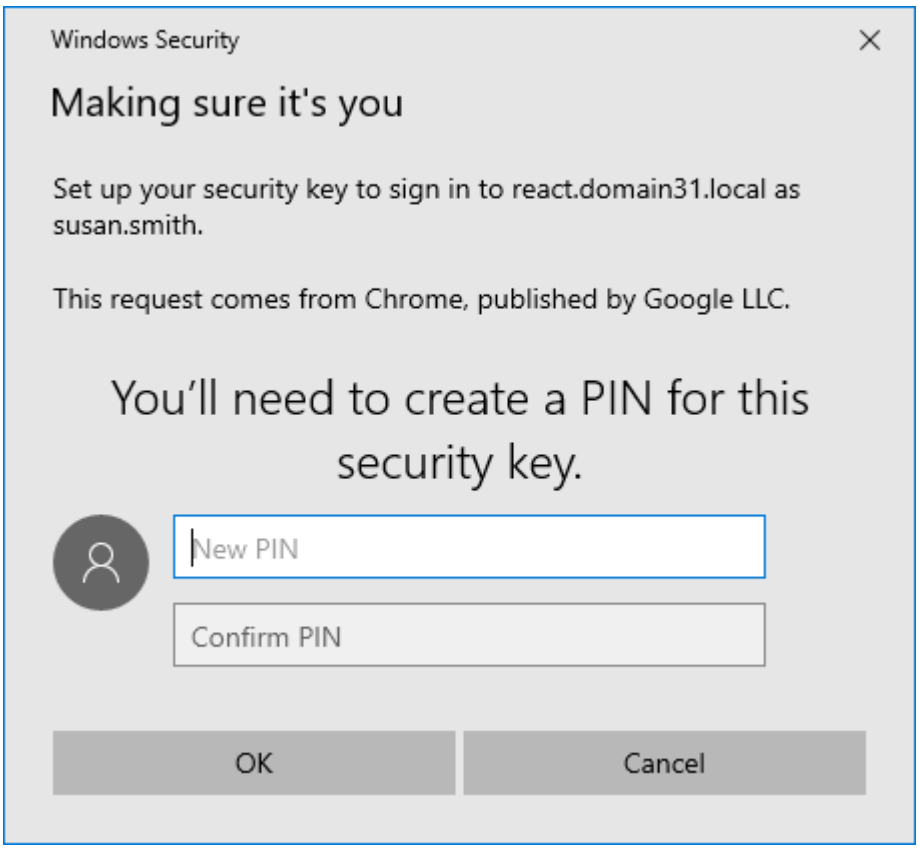
2. Type your registration code from the email or SMS you received, and click **Next**.

   Windows Security takes you through the registration process for your passkey. This process depends on the capabilities of your passkey, and is independent of MyID.

   For example, Windows Security prompts you to present your authenticator:

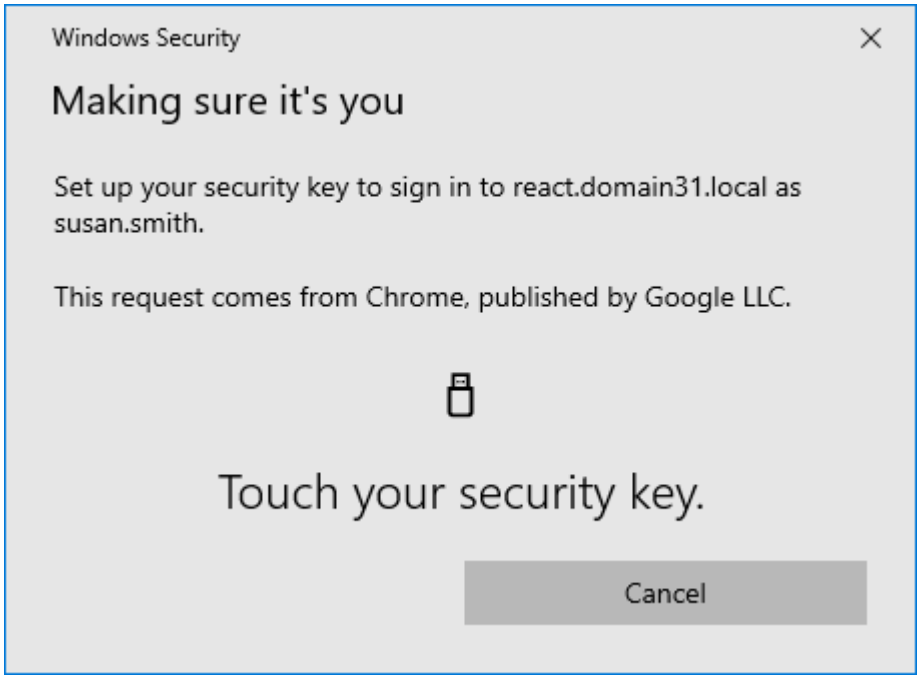
a. Present your authenticator.

You may be required to set up a PIN:



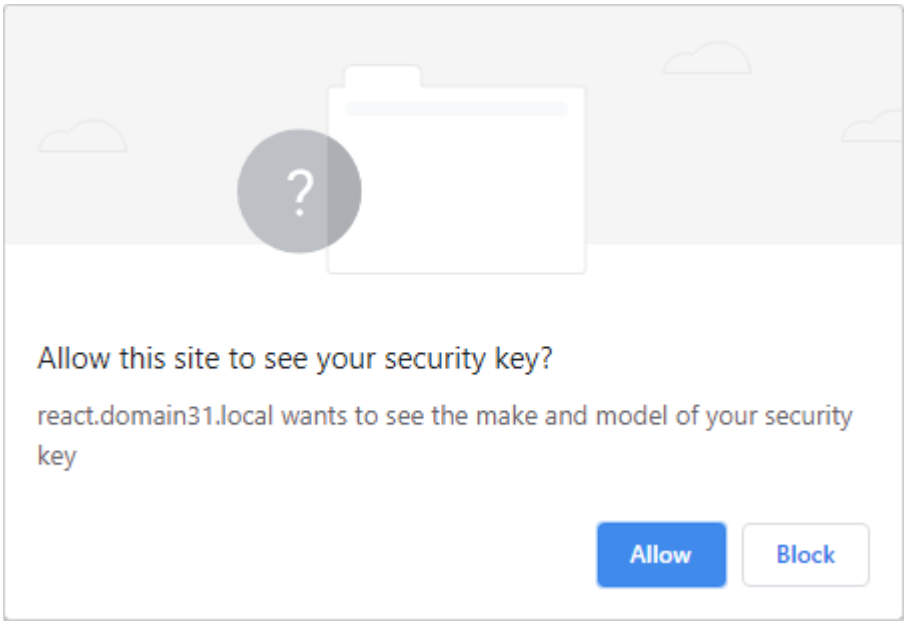
b. Type a **New PIN**, then confirm it, and click **OK**.

You may be required to provide additional authentication. For example, some passkeys require a PIN *and* for the user to touch the device physically for each authentication attempt; this provides an extra layer of security.



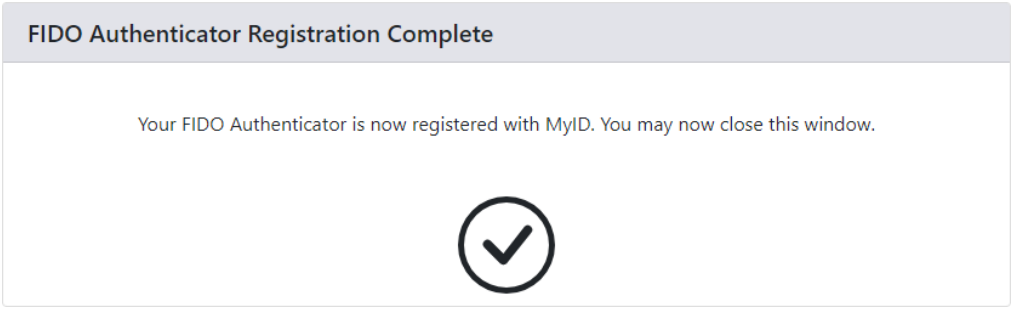

c. If your passkey requires it, touch the device.

Your browser may request that you allow the website to see the authenticator; for example, in Google Chrome:



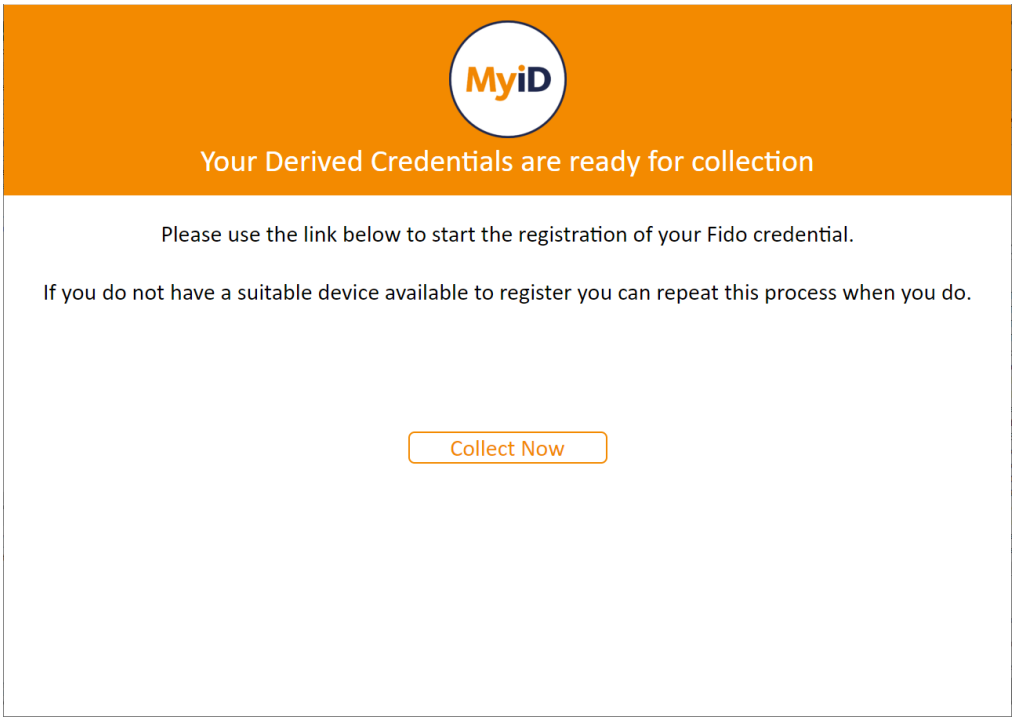

d. Click **Allow**.

When you have completed all the steps requested, your authenticator is registered with MyID, and is available for use. You can close the browser window.




### 6.2.2 Registering passkeys using the Self-Service Request Portal

If the credential profile used for the request had the **Immediate registration via Self-Service Request Portal** option set, and you requested the passkey using the Self-Service Request Portal, click **Collect Now** to begin the registration process.




Windows Security takes you through the registration process for your passkey. This process depends on the capabilities of your passkey, and is independent of MyID.

**Note:** The timeout for immediately collection is determined by the **FIDO Immediate Collect Timeout** option on the **PINs** tab of the **Security Settings** workflow. By default, the timeout is set for 120 seconds.

## 6.3 Viewing passkeys

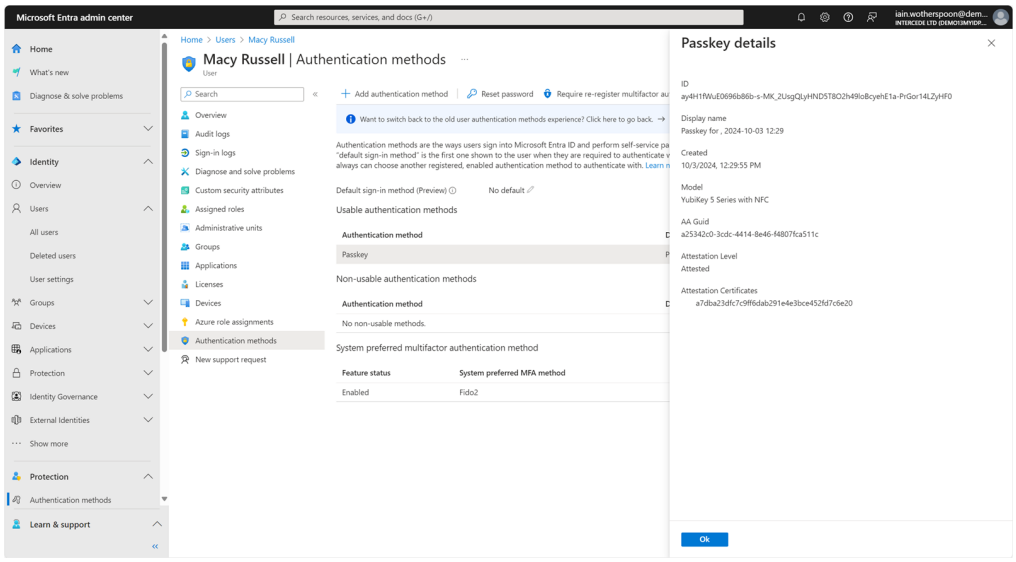
You can view the details of issued passkeys in Entra (if you are using Entra for authentication) or in MyID (whether you are using Entra or MyID CMS for authentication).

### 6.3.1 Viewing passkeys in Entra

To view the details of a passkey in Entra:

1. In the Microsoft Entra admin center, select the user who has been issued the passkey.
2. In the Authentication methods section, select the passkey.

   The details of the passkey are displayed.



### 6.3.2 Viewing passkeys in MyID

You can view the details of a passkey on the View Device screen in the MyID Operator Client.

To view the details for a passkey:

1. Log on to the MyID Operator Client.
2. Click the **Devices** category and search for the passkey you want to work with.
3. From the **Device Type** drop-down list in the search criteria, select one of the following options:
   - **FIDO Basic Assurance**
   - **FIDO High Assurance**

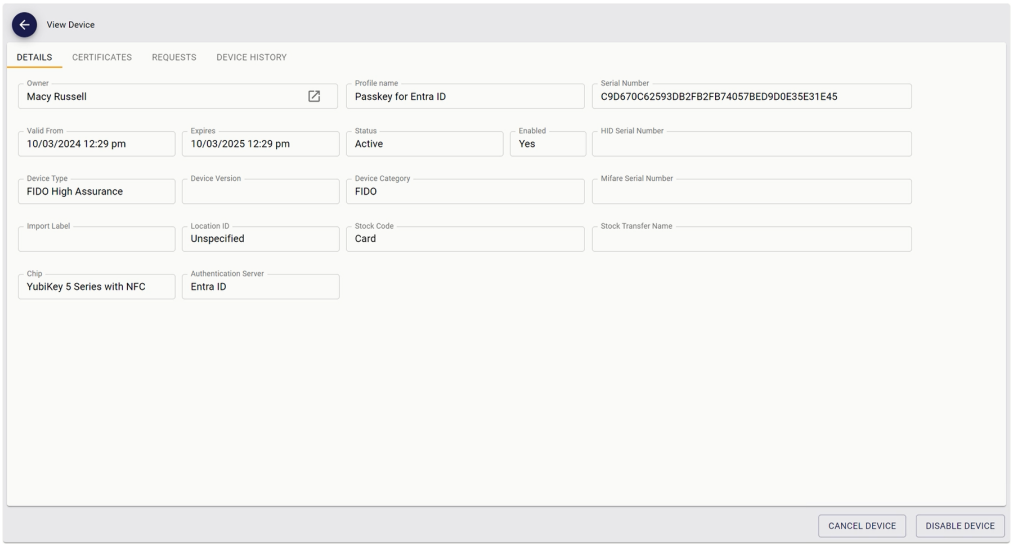
   You can use the standard Devices report, or the specific Passkeys report; see *Passkeys report* section in the ***MyID Operator Client*** guide for details.
4. Click **Search**.

   You can also view a device from any form that contains a link to the device.

   For example:
   - Click the item in the list on the **Devices** tab of the View Person form.
   - Click the link icon on the **Device Serial Number** field of the View Request form.

   See the *Searching for a device* section in the ***MyID Operator Client*** guide for more information about searching for devices.
5. Select the device.

   The View Device screen is displayed, containing the details of the passkey.



## 6.4 Enabling, disabling, and canceling passkeys

Once a passkey has been registered, you can use MyID to control its lifecycle.

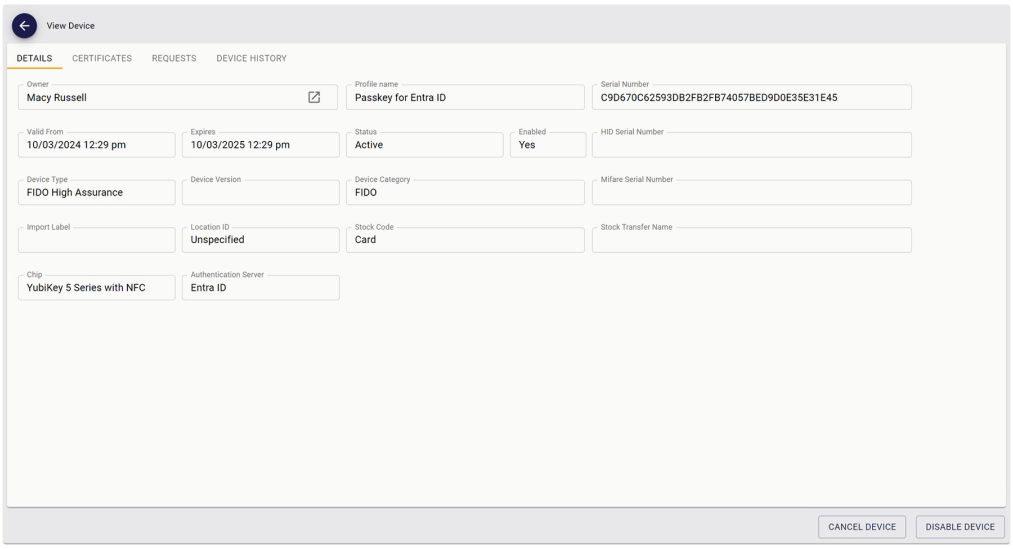
**Note:** This process is remote, and does not require physical access to the passkey. Additionally, this process does not affect the content of the authenticator; instead, it changes the status of the authenticator on the authentication service (either MyID CMS or Entra). If you want to change the content of the passkey, for example to carry out a complete reset, your authenticator manufacturer provides tools for this purpose.

### 6.4.1 Canceling passkeys

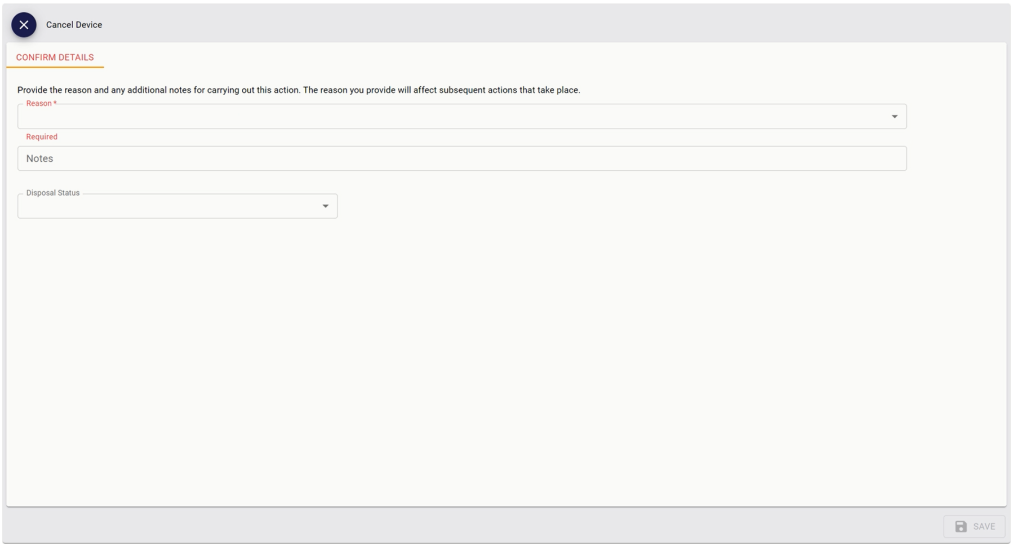
To cancel a passkey:

1. In the MyID Operator Client, search for the passkey you want to cancel.

   For example, you can find the passkey on the **Devices** tab of the View Person screen, in the **Devices** report, or in the **Passkeys** report. See section *6.3.2*, *Viewing passkeys in MyID*.




2. On the View Device screen, click **Cancel Device**.




3. From the **Reason** drop-down list, select the reason for the cancellation.

   As passkeys do not contain certificates issued by MyID, this option is for auditing purposes only.

4. Type any **Notes** on the cancellation in the provided box.

5. Click **Save**.

   The passkey is now canceled.

   If you are using Entra for authentication, you can log on to the Entra website as an administrator and confirm that the passkey is not listed for the user in the **Authentication methods** section; you can also check the user's **Audit logs** to confirm that the passkey has been deleted.

**Note:** You can also use the **Cancel Credential** workflow in MyID Desktop to cancel a passkey. See the *Canceling a credential* section in the ***Operator's Guide*** for details of using this workflow.

### 6.4.2 Enabling and disabling passkeys

You can disable a passkey to prevent it from being used temporarily. This does not affect the content of the device, but marks the passkey as disabled in the MyID database.

**Note:** You cannot disable a passkey on Entra. If you issued the passkey through Entra, the **Disable Device** button does not appear. If you carry out an action that has the effect of disabling a passkey (for example, disabling a person's user account) all Entra-authenticated passkeys on that account are canceled instead of disabled.

To enable or disable a passkey:

1. In the MyID Operator Client, search for the passkey you want to enable or disable.

   For example, you can find the passkey on the **Devices** tab of the View Person screen, in the **Devices** report, or in the **Passkeys** report. See section *6.3.2*, *Viewing passkeys in MyID*.
2. On the View Device screen, click **Disable Device** or **Enable Device**.
3. If you are disabling a passkey, select a **Reason** from the drop-down list.
4. Type any **Notes** you want to record.
5. Click **Save**.

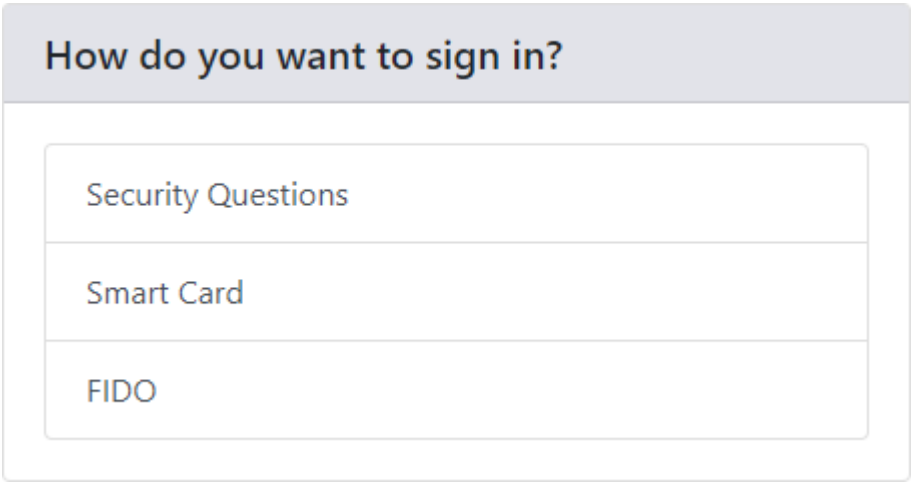
## 6.5 Signing in to MyID CMS with a passkey

If you have configured MyID to allow logon using FIDO passkeys (see section *2.6*, *Configuring MyID for logon with passkeys*) you can use a registered FIDO passkey to log on to the MyID Operator Client.

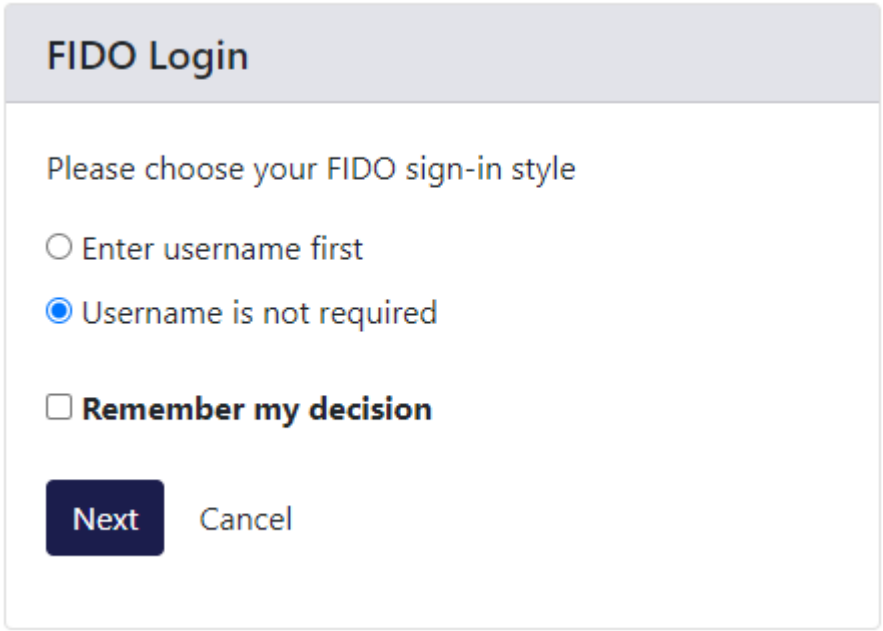
To log on to MyID using a FIDO passkey:

1. From the MyID Operator Client landing page, click **SIGN IN**.

   If more than one logon mechanism is configured for your system, you are prompted to select which one to use.



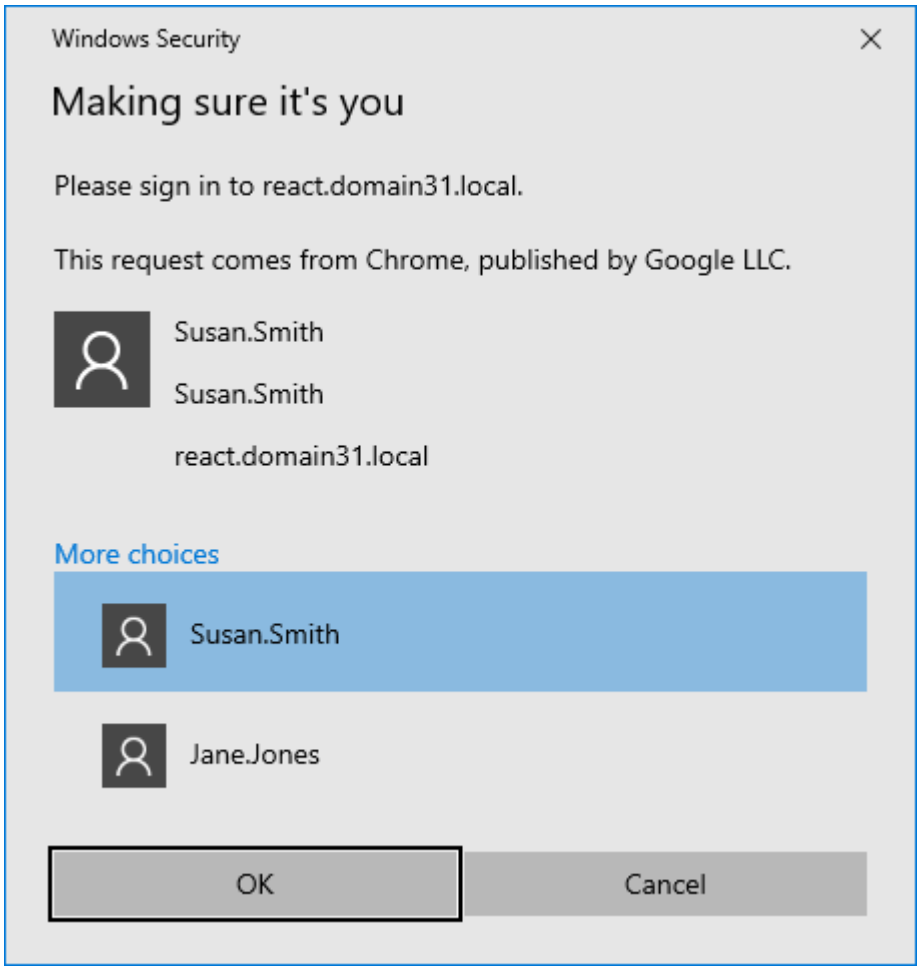

   Select **FIDO** from the list, and the FIDO Login screen appears:




2. Choose whether or not to provide a username:
   - **Enter username first**

     You must type your username when authenticating to MyID.
   - **Username is not required**

If your passkey supports it, and has been issued with a discoverable key for the user and the domain (for example, by issuing using a credential profile that has the **Require Client Side Discoverable Key** option set) you can opt not to provide a username. If there is more than one identity for the current domain on the passkey, the Windows Security dialog provides you with a list to select the appropriate one to use:




If you select **Remember my decision**, you will not be prompted again when using this browser under this user account on this PC. If you subsequently change your mind, you can click **Cancel** on a FIDO system authentication dialog box, or delete the cookies stored in your browser from the MyID website.

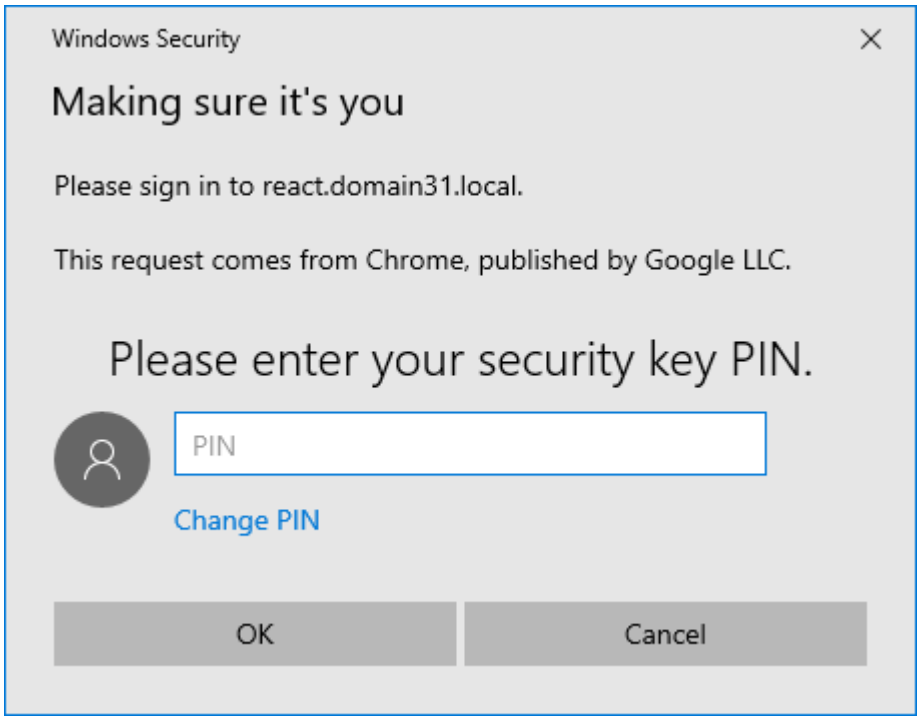
3. Complete your FIDO authentication.

   **Note:** The specifics of the process depend on the capabilities of your passkey, your selected FIDO logon style (username or no username) and how your credential profile is set up for user verification.

   For example:

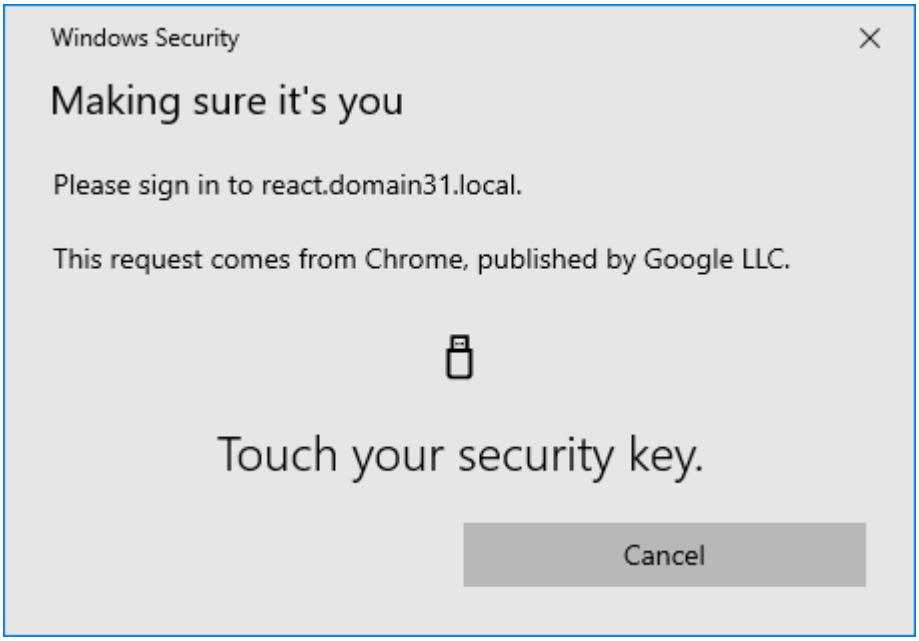
   a. Type your **Username** and click **Next**.

      The Windows Security dialog appears, requesting your PIN:



b. Enter your PIN and click **OK**.

The Windows Security dialog requests that you touch your authenticator:




c. Touch your authenticator.

You can now use the MyID Operator Client.

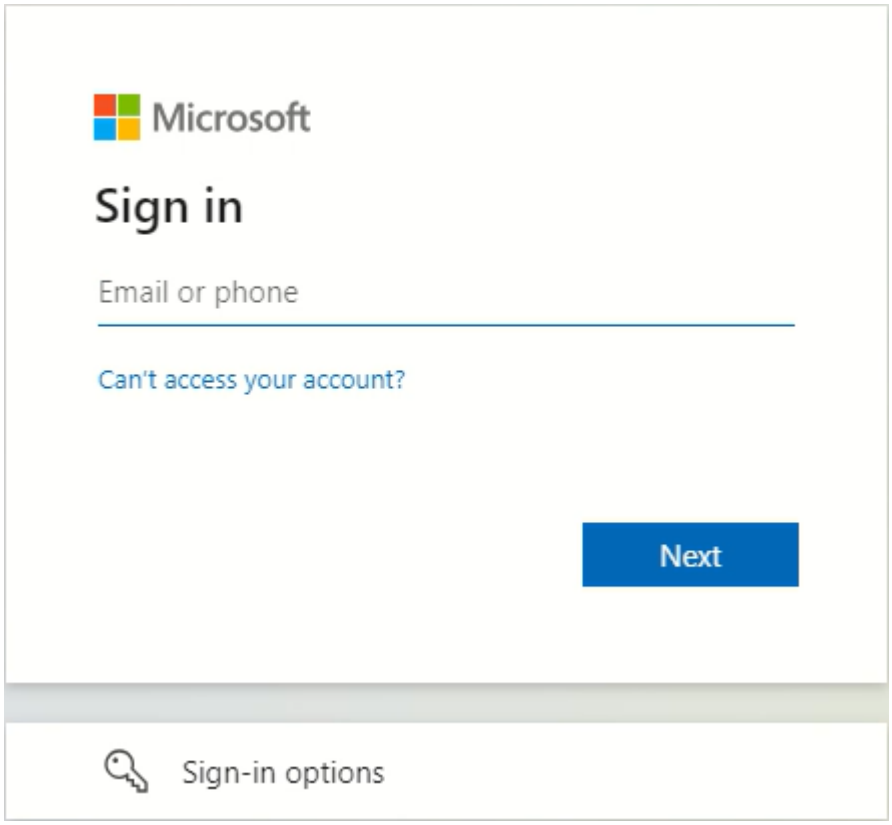
**Note:** If the list of features available in the MyID Operator Client does not match what you expect, check that the logon mechanisms have been set up correctly for your roles; see section *2.6.2*, *Setting up FIDO logon mechanisms*.

## 6.6 Signing in to Microsoft using a passkey

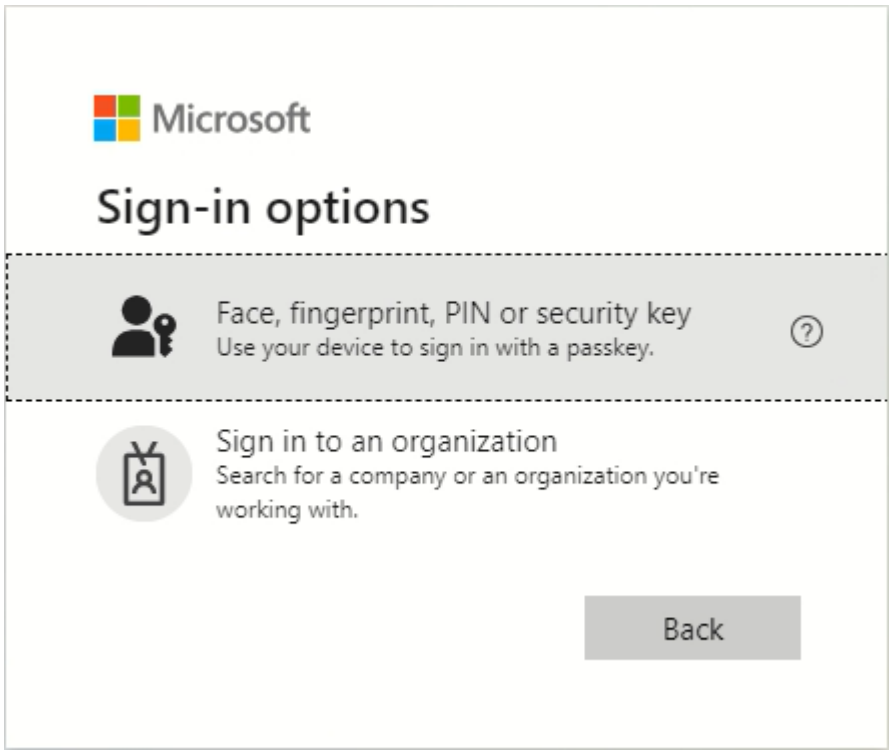
Once you have registered a passkey on your device with Entra, you can use it to sign in to Microsoft.

To use a passkey to sign in to Microsoft:

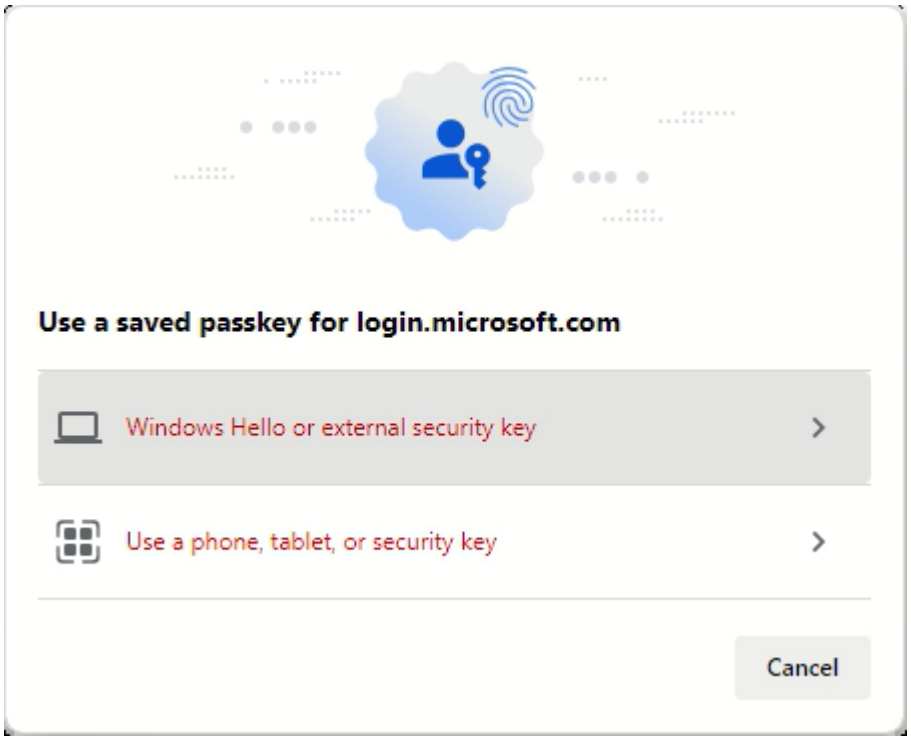
1. Navigate to the Microsoft sign in page.



2. Click **Sign-in options**.

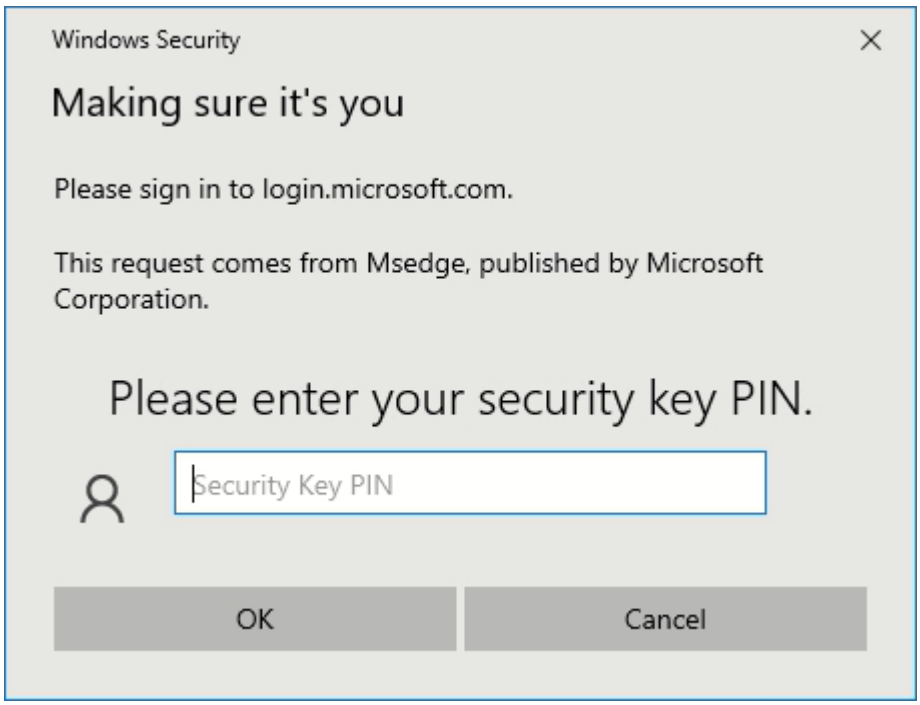



3. On the Sign-in options screen, select **Face, fingerprint, PIN or security key**. Your device opens a security window.



4. Select **Windows Hello or external security key**.

   The Windows Security system checks your credentials.




5. Follow the on-screen instructions; for example, enter your PIN and touch the device.

   You can choose whether to stay signed in; click **Yes** to stay signed in, or **No** to sign out after you leave the site.

   You are now logged on to the Microsoft website securely using your passkey.

# 7 Troubleshooting

This section contains troubleshooting information and frequently asked questions related to working with passkeys.

If you experience any issues when working with passkeys, check the **Audit Reporting** workflow in MyID Desktop for details of the error that has occurred.

If an error number is displayed; for example:

```
OA10084 – An error occurred while trying to retrieve an access token from
an external source.
```

you can look up this error code in the *MyID Operator Client error codes* section of the ***Error Code Reference*** guide.

If you are issuing passkeys through Entra, you are recommended to review Microsoft's list of known issues:

***learn.microsoft.com/en-us/entra/identity/authentication/how-to-enable-passkey-fido2#known-issues***

- **I requested a FIDO token, but there are no notifications**

  Check your email SMTP server settings in the **External Systems** workflow; see the *Setting up email* section in the ***Advanced Configuration Guide***.

  Check that the person has an email address stored in MyID. If you are using SMS to distribute the registration codes, check that the person also has a cell/mobile number set up.

  In Windows Services, check that both the MyID Notifications Service and eCertificate Services Server are running.

- **I requested a FIDO token and one notification arrives instantly but the other notification takes longer to arrive**

  The two notifications are sent by different processes on the MyID server (one through the MyID Notifications Service and the other through the eCertificate Services Server) so the notifications may arrive on slightly different schedules depending on the polling time of the services.

- **The link in the registration email does not work**

  If the link does not start with `https://<your server name>` check that the **URL path** option is set; see section *2.1*, *Setting the configuration options*.

  **Note:** This link *must* be an https address.

- **My FIDO registration code is not accepted**

  Make sure that the **Allow Logon Codes** option is set; see section *2.1*, *Setting the configuration options*.

  Make sure that the person has permission to the **Register FIDO Security Key** option in Edit Roles, and that the role has the Password logon mechanism; see section *2.5*, *Configuring roles for registering passkeys*.

- **My FIDO registration code was accepted, but I get error OA10009**

  Check that your browser and authenticator support FIDO2 Web Authentication (WebAuthn) standard.

  For more information about browsers, operating systems, and authenticators that support this, see:

  ***fidoalliance.org/fido2/fido2-web-authentication-webauthn/***

  Check that the FIDO credential profile is compatible with the type of passkey device you want to use; not all devices support all FIDO features.

- **When attempting to issue a passkey through Entra, I get error OA10009**

  This may occur if you have not configured the **Authentication Server** option in the credential profile to the name of the external system you created for Entra.

  This may also occur if the external system you created for Entra has the wrong **API Location**.

- **When registering, I get error OA10017 – a problem accessing the FIDO metadata**

  Make sure that the web server has access to the Internet, and there is no firewall preventing access to the FIDO metadata service; for example, make sure you can access the `mds.fidoalliance.org` domain. (This domain is controlled by the FIDO Alliance, and may be subject to change.)

  See section *2.3*, *Setting up the FIDO metadata*.

- **I want to supply my own metadata, as the authenticator I used is not on the FIDO metadata service, or I want to restrict issuance to a specific passkey**

  If you want to use your own file-based FIDO metadata repository, follow the instructions in section *2.3.1*, *Setting up a local metadata repository*.

- **When logging on to the MyID Operator Client, I do not get the option to select FIDO as a means of logging on**

  Make sure that you have enabled at least one of the FIDO logon mechanisms; see section *2.6*, *Configuring MyID for logon with passkeys*.

  Check the `appsettings.json` file for the web.oauth2 service; by default, this is:

  ```
  C:\Program Files\Intercede\MyID\web.oauth2\appsettings.json
  ```

  Check that the `EnableFido2LoginBasicAssurance` and `EnableFido2LoginHighAssurance` options have not been set to false for the MyID Operator Client (`myid.operatorclient`).

- **I cannot log on to MyID with my passkey**

  Check the credential profile – the **MyID Logon** option in the `Services` section must be enabled to allow MyID logon.

  Check that the person and device are enabled, and that the device has not expired.

  The **Audit Reporting** and **System Events** workflows may provide additional information.

  You can also check the `AuthenticationAudits` table in the authentication database); see the *Reporting on the authentication database* section in the ***MyID Authentication Guide*** for details.

- **When trying to log on with FIDO, there is an error complaining about the domain or origin**

  FIDO tokens are domain locked to the domain that registered them; that is, if a website at `https://myserverdomain` registered the passkey, that passkey can be used only to authenticate at websites that are also at `https://myserverdomain`. However, the same passkey can hold FIDO credentials for other systems that MyID does not know about, enabling a user to use that passkey for many systems; MyID will ignore these other FIDO credentials.

  Therefore it is important to not change the server domain of the MyID system, as doing so will render already registered FIDO credentials unusable; if this happens, you must request and register new FIDO credentials.

  **Note:** It is the URL the client sees that is important; this may be the URL of the load balancer or reverse proxy they access rather than the URL of the actual MyID web server.

  There is special consideration if alternative web servers are used for a standalone MyID authentication service (for externally facing systems such as ADFS to authenticate to) but the passkey is registered on a different MyID web server by web.oauth2. In this situation, you must set up a load balancer or proxy so the same domain is accessed in both cases and routed to the appropriate servers.

  **Note:** MyID now supports multiple origins, where sub-domains of a registrable domain can also be authenticated; see section *2.4.3*, *Multiple origins*.

- **When trying to log on with FIDO I get the error HTTP 431 Request Header Fields Too Large**

  Your passkey has too many credentials on it, which is causing the combined length of the credential IDs to exceed the HTTP header size restriction; you are recommended to cancel any older unwanted FIDO credentials for that user.

- **I cannot log on using my older FIDO credentials**

  This is related to the HTTP 431 Request Header Fields Too Large error. When the combined length of credential IDs is too large and runs the risk of exceeding the HTTP header size restriction, the older tokens are ignored.

- **Why are there two logon mechanisms – FIDO Basic Assurance and FIDO High Assurance?**

  This provides flexibility. You may want to issue one-factor authenticators (basic assurance) for logging on to some external systems, but only allow two-factor authenticators (high assurance) for logging on to MyID.

- **I registered two FIDO credentials to the same device, but MyID shows them as two separate devices – why?**

  FIDO has privacy built in that prevents a system from identifying the authenticator to which the credential is issued; this means that each registered FIDO credential has its own device record.

  However, if you are using enterprise attestation, this can provide a link between multiple passkeys on the same device using passkey serial numbers based on the physical device's serial number.

- **Does this mean each FIDO credential uses a MyID device license, even if they are on the same physical device?**

  Yes, MyID tracks device license usage based on issued credentials, not physical devices.

- **I get database errors when registering or authenticating a passkey**

  Make sure that the authentication database is set up correctly. Ensure that the authentication `.udl` file (by default, `MyIDAuth.udl`) in the Windows `System32` folder of the MyID application server is pointing to the authentication database.

- **I get HTTP Error 500.30**

  If you see an error similar to:

  ```
  HTTP Error 500.30 - ANCM In-Process Start Failure
  ```

  Check that your `appsettings.Production.json` file is valid.

  Note especially that copying code samples from a browser may include hard spaces, which cause the JSON file to be invalid.

  To assist in tracking down the problem, you can use the Windows Event Viewer. Check the **Windows Logs > Application** section for errors; you may find an error from the .NET Runtime source that contains information similar to:

  ```
  Exception Info: System.FormatException: Could not parse the JSON file.
  ---> System.Text.Json.JsonReaderException: '"' is invalid after a value. Expected either ',', '}', or ']'. LineNumber: 13 | BytePositionInLine: 6.
  ```

  which could be caused by a missing comma at the end of a line.

  An error similar to:

  ```
  Exception Info: System.FormatException: Could not parse the JSON file.
  ---> System.Text.Json.JsonReaderException: '0xC2' is an invalid start of a property name. Expected a '"'. LineNumber: 7 | BytePositionInLine: 0.
  ```

  is caused by a hard (non-breaking) space copied from a web browser, which is not supported in JSON.

  **Note:** Some JSON files used by MyID contain comment lines beginning with double slashes `//` – these comments are not supported by the JSON format, so the JSON files will fail validation if you attempt to use external JSON validation tools. However, these comments *are* supported in the JSON implementation provided by asp.net.core, and so are valid in the context of MyID.

- **I get errors relating to attestation when registering a GoTrust device**

  There have been issues noticed when registering GoTrust Idem Key device due to a problem with the GoTrust root certificate. Contact GoTrust technical support for assistance.

- **Error code OA10007: Your OTP has been entered incorrectly, is locked, has expired, or you do not have permission to perform this operation. Please try again.**

  Solution: The user must have `PasswordUser` role. You can configure the SSRP mappings in the `myid.json` file to add this role automatically; for example:

```
{
  "From":"",
  "To":"Roles",
  "Static":"<role name='Derived Credential Owner' scope='1'/><role
name='Cardholder' scope='1'/><role name='PasswordUser' scope='1'/>"
},
```

  See the *Configuring the Self-Service Request Portal for external identity providers* section in the ***Derived Credentials Self-Service Request Portal*** guide for details of assigning roles.

- **Error when using Hyper-V client running Windows 11**

  You may experience errors if you attempt to use a Hyper-V client to issue passkeys; for example:

  `OA10088 – The MyID client service encountered an issue while attempting to register your device with an external source`

  You cannot use a Hyper-V client to issue passkeys.