



MyID Enterprise

Version 12.14

Derived Credentials Self-Service Request Portal

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

Copyright

© 2001-2025 Intercede Limited. All rights reserved.

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

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

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

Trademarks

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

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

Licenses

This software includes packages provided under a variety of licenses. The *About the documentation* page in the HTML version of the MyID CMS documentation, available with the MyID CMS software or on the Intercede customer portal website, contains a full list.

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

Derived Credentials Self-Service Request Portal	1
Copyright	2
Conventions used in this document	3
Contents	4
1 Introduction	6
1.1 Prerequisites	6
1.2 SSRP overview	7
1.2.1 Using the SSRP on a mobile device	7
1.3 Impact on SP800-157 compliance	8
1.4 What information is stored in MyID when I request a Derived Credential?	8
2 Configuring the Self-Service Request Portal	9
2.1 Installing the Self-Service Request Portal	9
2.2 Setting up COM+ proxies	9
2.3 Setting up SSL/TLS on the SSRP	10
2.3.1 Web applications	10
2.3.2 SSL certificates	10
2.3.3 Using TLS 1.3	11
2.4 MyID configuration options	11
2.4.1 Setting the credential check period	14
2.4.2 Determining which cards are available for derived credentials	15
2.4.3 Configuring certificate OIDs checked on PIV cards	16
2.5 Configuring email notifications	16
2.5.1 Setting up email	16
2.5.2 Editing the request email template	17
2.5.3 Editing the cancellation email template	18
2.5.4 Obtaining the email address	19
2.5.5 Requiring an email address	19
2.5.6 Updating the email address	21
2.6 Granting access to the workflows	21
2.6.1 Role permissions	21
2.7 Setting up the certificate checks	21
2.8 Setting up the credential profiles for derived credentials	21
2.8.1 Creating an Identity Agent credential profile	22
2.8.2 Creating a mobile identity document credential profile	24
2.8.3 Creating a VSC credential profile	26
2.8.4 Creating a Windows Hello credential profile	27
2.8.5 Creating a FIDO authenticator credential profile	29
2.8.6 Creating a credential profile for other devices	33
2.8.7 Credential profile restrictions	34
2.8.8 Configuring the available credential profiles	35
2.8.9 Mapping certificates to roles and credential profiles	35
2.8.10 Restricting based on the certificate authority path	36
2.8.11 Verifying certificates	36

2.8.12 Configuration file format	36
2.9 Lifecycle management	38
2.10 Request validation	39
3 Requesting a Derived Credential	40
3.1 DN format	40
4 Handling revocation	41
5 Multiple credential handling	42
5.1 Users with multiple DNs	42
5.2 Users with multiple devices	42
5.3 Archived private key handling	42
6 Customizing the Self-Service Request Portal	43
6.1 Customizing the terminology	43
6.1.1 Providing information on the issuance process	45
6.2 Customizing locations	45
6.3 Customizing the properties file	46
6.3.1 Editing the provider properties	46
6.3.2 Customizing the Self-Service App links	47
7 External identity providers	48
7.1 Configuring multiple authentication provider types	48
7.2 Configuring your external identity provider	49
7.3 Configuring the Self-Service Request Portal for external identity providers	51
7.3.1 Encrypting the client secret	58
7.4 Sample configuration for Entra	59
7.4.1 Configuring the file for your own settings	59
7.4.2 Setting up redirect URIs	59
7.4.3 Example myid.production.json file for Microsoft Entra	60
8 Error codes and logging	62
8.1 Error code reference	63
8.1.1 SSL authentication error	66
8.2 Troubleshooting mobile device collection	66
9 Mutual SSL behavior	67
9.1 Caching credentials	67
9.2 Persisting credentials	67
9.3 Certificate selection hidden from user	67
9.4 Smart card not registered	67

1 Introduction

The Derived Credentials Self-Service Request Portal (SSRP) allows you to request a MyID® derived credential for your smart card, USB token, smartphone, FIDO authenticator, or Windows PC (Microsoft VSC or Windows Hello for Business); this derived credential is based on credentials you already have from another identity provider.

The Self-Service Request Portal supports the following types of identity provider:

- Client certificate from a PIV card – you log in with your PIV card, and SSRP generates derived credentials based on the client certificate stored on the card. This is the default. If this is the only authentication type that you want, you do not need to configure external identity providers.
- OpenID Connect – you authenticate to an external identity provider, and SSRP generates derived credentials based on the claims returned by the external system.

You can configure SSRP for multiple OpenID Connect providers; SSRP provides a choice of providers to the user when they access the SSRP website. You can also configure SSRP for one or more OpenID Connect providers in addition to the PIV card provider.

The SSRP is a website that does not require any additional software to be installed on the client PC. It works with the following browsers on Microsoft Windows PCs:

- Microsoft Edge
- Google Chrome

If you would like to use the SSRP with a different web browser, contact customer support quoting reference SUP-321.

You can request derived credentials from existing credentials that were issued by the current MyID system, or from credentials that were issued by external systems.

1.1 Prerequisites

SSRP requires the following:

- .NET
The web server on which you install the SSRP must have .NET 4.8 or later installed.
- MyID Identity Agent app
For mobile-based derived credentials, you must have the MyID Identity Agent app installed on your mobile device. You can configure SSRP to allow you to download the Identity Agent onto your mobile device.
See the [Mobile Identity Management](#) guide for system requirements and details of configuring your system for mobile identities.
You can also issued mobile-based derived credentials to the MyID Authenticator app. See the [Mobile Authentication](#) guide for details.
- Trusted Platform Module
For Microsoft VSC-based derived credentials, you must have a PC with a Trusted Platform Module. See the [Microsoft VSC Integration Guide](#) for system requirements and details of configuring your system for Microsoft VSCs.

1.2 SSRP overview

Derived Credentials are certificates issued to a person following authentication with a trusted credential that has already been issued to them from a different source. The 'trust' in the certificate used to authenticate is carried forward to the new certificates.

The request process is as follows:

A person has an already-issued trusted credential; typically, a smart card containing certificates that can be used for authentication, or an account on an external system.

For smart cards, the card may have been issued by your MyID system, or by another system that issues smart cards. The certificates on the card must be in date and capable of performing Client Authentication. On their PC, the person will visit the SSRP website and insert their card into a card reader.

Alternatively, you can configure SSRP to use a person's credentials from an external identity provider that uses OpenID; the person visits the SSRP website, clicks through to the external identity provider, authenticates to the external system, and SSRP generates a derived credential based on the claims returned from the external system.

The SSRP checks the server to see what types of derived credentials are available to the cardholder, based on a variety of configurable options such as whether their original credential was issued by the current system, the DN of the originating CA of their certificates, and the properties of the certificates on their card; if more than one type of credential is available, the SSRP presents a list. The cardholder selects the type of derived credential, and is then presented with instructions on how to collect the derived credential to their smart card, USB token, mobile device, VSC, or Windows Hello for Business.

If the person's user account already exists within your MyID system, the derived credential is added to their user account. If the original credentials were issued by an external system, and the configuration of your system permits it, the unknown person's details are taken from their original credentials and added to MyID; optionally, MyID checks the LDAP for the user and incorporates those details, too. Links to LDAP are based on the person's DN and UPN (for client certificate-based derived credentials) or a configurable mapping from the claims (for OpenID Connect based derived credentials).

Note: It is possible for a MyID system to be connected to multiple directories. If the system is configured this way, it may not be possible for MyID to identify group assignments for users not in the primary directory. When this occurs, the user is placed in a group matching their LDAP group (or Agency for PIV), within the Derived Credentials group. This does not affect the issued credential, just the location of the user account within the MyID group hierarchy.

1.2.1 Using the SSRP on a mobile device

You can use the SSRP on a mobile (iOS or Android) device if you have a mobile card reader. You visit the SSRP website on the mobile device, use the attached mobile card reader to present your existing credential, and the SSRP displays a link that allows you to collect the derived credential directly on the mobile device. To configure this feature, you must set up the **Collection Instructions** for the credential profile to specify which app you want to use to collect the derived credential; see section 2.8, [Setting up the credential profiles for derived credentials](#).

1.3 Impact on SP800-157 compliance

SSRP is designed to provide a more flexible and easy-to-use method of requesting and issuing derived credentials. It also provides capabilities to synchronize user information from a directory, such as Microsoft Active Directory or data imported using a MyID API. This will impact the information available to be used on any client certificate-based derived credentials.

For example, the distinguished name of the certificates issued as derived credentials will be different to the certificate used for authentication during the request in cases where you have used a directory as the central data source and the DN in the directory does not match the DN on the original PIV card.

SSRP can import data only from the original PIV Authentication certificate, and data that is synchronized from LDAP. The Self-Service Kiosk has additional capabilities over SSRP as it can access more data from the PIV card (for example, read biometric data from the card). Note that SP800-157 LOA3 credentials do not require biometric authentication.

A 7 day revocation check is performed.

1.4 What information is stored in MyID when I request a Derived Credential?

For PIV card-based derived credentials, MyID extracts the DN and UPN from the certificate. The certificate owner's name is deduced from the DN. Additionally, the certificate itself is stored for a configurable period to allow a follow-up validity check on it; by default, this is 7 days.

In addition to this, if LDAP synchronization is enabled, MyID extracts data from the LDAP as configured in the `LDAPMapping` table. This may include phone numbers and email addresses if so configured.

For external identity providers, MyID extracts information from the claims provided by the external system. You have full control over mapping these claims to user attributes within MyID.

2 Configuring the Self-Service Request Portal

This chapter contains information on configuring the SSRP, including:

- Installing the Self-Service Request Portal.
See section [2.1, *Installing the Self-Service Request Portal*](#).
- Setting Up COM+ proxies.
See section [2.2, *Setting up COM+ proxies*](#).
- Setting up SSL/TLS.
See section [2.3, *Setting up SSL/TLS on the SSRP*](#).
- Setting the MyID configuration options.
See section [2.4, *MyID configuration options*](#).
- Granting access to the workflows.
See section [2.6, *Granting access to the workflows*](#).
- Setting up certificate checks.
See section [2.7, *Setting up the certificate checks*](#).
- Setting up credential profiles.
See section [2.8, *Setting up the credential profiles for derived credentials*](#).
- Lifecycle management of derived credentials.
See section [2.9, *Lifecycle management*](#).
- Validating requests.
See section [2.10, *Request validation*](#).

2.1 Installing the Self-Service Request Portal

To install SSRP, when you install MyID, select the following option on the Server Roles and Features screen:

- **Self-Service Request Portal Web Service**

This installs the SSRP website.

2.2 Setting up COM+ proxies

If you have installed the SSRP on a separate server to the MyID application server, you must install the MyID COM+ proxies to allow the SSRP site to communicate with the components on the MyID application server.

To do this, you need the .msi files in the `Components\Export` folder on the MyID application server. By default, this is:

```
C:\Program Files\Intercede\MyID\Components\Export
```

To run the COM+ proxy installers, either:

- From the MyID web server, browse to a share on the MyID application server and run the .msi installers directly. For example, browse to:

```
\\<server>\C$\Program Files\Intercede\MyID\Components\Export
```

where `<server>` is the name of your MyID application server and `C$` is a share of the root of the `C:` drive. Run the `.msi` files directly.

Note: If you experience any problems, make sure you have added the application server to the list of Trusted Sites on the web server.

or:

- Copy the `.msi` files to the MyID web server and run the installers from there.

Note: If you are using multiple servers for your web services in conjunction with a load balancer, you must ensure that you set up session affinity on your servers. See also the *Reverse proxies and load balancing* section in the [Web Service Architecture](#) guide.

2.3 Setting up SSL/TLS on the SSRP

2.3.1 Web applications

The SSRP system comprises the following web applications:

- `Start` – the launch page that allows you to select from multiple identity providers.
- `StartPage` – the launch page that allows you to use only a client certificate on a PIV card as the source for the derived credential.
- `SSRP` – the web application that carries out secure requests for client certificate-based derived credentials.
- `SSRPOID` – a web application you can configure for OpenID authentication for derived credentials based on information from an external system.

You must set up the `Start`, `StartPage`, `SSRP`, and `SSRPOID` web applications to require 1-way SSL/TLS.

If you are using PIV card-based derived credentials, you must also set up the `SSRP` web application to require 2-way SSL/TLS. This web application verifies the cardholder's request and initiates the issuance of the client certificate-based derived credential. Make sure the **Client certificates** option in IIS is set to **Accept**; this is required if you are using the `Start` launch page.

If you are using only the `StartPage` launch page, and do not require OpenID as an authentication option, you can set the **Client certificates** option in IIS to **Require**.

2.3.2 SSL certificates

For client certificate-based derived credentials, it is important that the IIS server has a certificate in its trusted root that matches a certificate in the user's SSL certificate chain.

The user's computer must also have a certificate in the trusted root CA that matches a certificate in the server's SSL certificate chain.

Certificates that have expired or are not trusted by the IIS server are not eligible for use and may well be hidden by the browser.

2.3.3 Using TLS 1.3

By default, TLS 1.3 is enabled on Windows Server 2022. To use TLS 1.3 with the Self-Service Request Portal, you must modify the https port bindings to enable client certificate negotiation. You cannot do this through the IIS user interface, but you can use a PowerShell script instead.

To enable the use of TLS 1.3 for the SSRP:

1. Ensure TLS 1.3 is enabled for the web server:
 - a. In Internet Information Services (IIS) Manager, in the **Connections** pane, expand the server name, then **Sites**, then select the website used for SSRP; by default, this is **Default Web Site**.
 - b. Right-click the website, then from the pop-up menu select **Edit Bindings**.
 - c. In the Site Bindings dialog, select **https**.
 - d. Click **Edit**.
 - e. Make sure the **Disable TLS 1.3 over TCP** option is not selected.
 - f. Click **OK**, then click **Close**.
2. Take a note of the server TLS certificate thumbprint:
 - a. In IIS, select the server on which SSRP is installed, then double-click **Server Certificates**.
 - b. Double-click the certificate used for TLS authentication.
 - c. Select the **Details** tab.
You can view the **Thumbprint** field here.
3. Run the following PowerShell script, substituting in the thumbprint in the `serverTlsCertThumbprint` variable:

```
$port = 443
$serverTlsCertThumbprint = '[thumbprint]'
$iisBindingAppId = New-Guid
Start-Process -FilePath "netsh" -ArgumentList @("http", "del", -join
("sslcert", " 0.0.0.0:", $port)) -Wait
Start-Process -FilePath "netsh" -ArgumentList @("http", "add", "sslcert",
-join("ipport=", "0.0.0.0:", $port), -join("certhash=",
$serverTlsCertThumbprint), -join("appid={", $iisBindingAppId, "}"),
"certstorename=MY", "clientcertnegotiation=Enable") -Wait
```

Note: Any subsequent changes to the https binding through the IIS user interface reverts this change. If this happens, run through the above steps again to re-enable TLS 1.3 for SSRP.

2.4 MyID configuration options

SSRP uses the following MyID configuration options when working with client certificate-based derived credentials. These options do not affect OpenID-based derived credentials.

- **Allow derived credential requests to create accounts**

This option appears on the **Issuance Processes** page of the **Operation Settings** workflow

If this setting is referred to in the audit trail, it appears using the internal name `DERIVED CREDENTIALS ALLOW IMPORT USERS`.

Must be set to **Yes** to allow SSRP to issue a derived credential to a cardholder whose original credential was issued by a different system. The unknown user is added to MyID.

When this option is set to **Yes**, when a trusted credential is used to import a new user into MyID using SSRP, MyID creates a new group in which the user will be placed if the required group does not already exist. For PIV certificates that contain a FASC-N, the Agency code is used in the group name (constructed as `Agency - <agencyCode>` – for example, `Agency - 0001`). For certificates that do not contain a FASC-N, the organizational unit identified in the subject distinguished name is used. In each case, MyID attempts to identify existing groups using the respective identifiers.

If a matching group is found and the group is associated with an LDAP configuration, the LDAP configuration is also used for the imported user.

If a matching group is not found, but the subject distinguished name in the trusted credential conforms with the distinguished name format used in LDAP v3 directories, then MyID attempts to determine which LDAP the user belongs to. If MyID is unable to determine an appropriate LDAP, either because the subject DN does not match a configured LDAP or the DN is not LDAP v3 compliant, the Default ADS LDAP connection will be used (if configured).

Finally, if an LDAP connection was identified, the imported user is associated with the LDAP. If MyID has been unable to determine a suitable LDAP connection by means described, the imported user will not be associated with an LDAP.

- **Assign unmatched new accounts to default directory**

This option appears on the **LDAP** page of the **Operation Settings** workflow.

When a new user account is created in MyID, the user OU may not be able to be matched to a directory OU that is matched to a MyID group; if there is a match then the user is linked to that group in that directory, but if there is not, you can set this option to **Yes** to link the account to the default directory registered with MyID.

- **Synchronize new accounts with directory**

This option appears on the **LDAP** page of the **Operation Settings** workflow.

If this setting is referred to in the audit trail, it appears using the internal name `DERIVED CREDENTIALS SYNC NEW USERS WITH LDAP`.

SSRP does not import the user's email address from a PIV card, since the email address is not present on the PIV Authentication certificate. If you want to issue (email) signing/encryption certificates as derived credentials, and you have the appropriate data in your LDAP directory, you can enable the **Synchronize new accounts with directory** feature so that additional data, including the email address, is imported from the directory

If this option is set to **Yes**, immediately after importing an unknown user MyID will attempt to pull extended details for that user from LDAP. A match will first be attempted using the DN of the certificate used to make the request. If no match is found, and the certificate contains a UPN, a second attempt will be made to match against the UPN. If both of these fail to match, no further data will be imported for the account.

This approach allows the system to consolidate users with multiple DNs but a common UPN into a single account, making collection easier.

Note: If you set the **Synchronize new accounts with directory** option to **Yes**, you must set the **Disable on removal from directory** option (on the **LDAP** page of the **Operation Settings** workflow) to **No**; if you do not do this, newly-created accounts that do not match a directory entry will become disabled, preventing the issuance of a derived credential.

Note: If this feature is enabled, and the user is matched against the UPN, the user's DN will be imported from the directory. If the DN in the directory does not match the DN on the original PIV card, this can cause the PIV derived credential to be issued with the DN from the directory, which may differ from the DN on the original PIV Authentication certificate.

- **Update email address from derivation**

This option appears on the **Certificates** page of the **Operation Settings** workflow.

Set this option to **Yes** to update the MyID record for the derived credential owner with the email address obtained from the certificate used for derivation.

The default is **No**.

- **Limit derived credential lifetime to deriving credential**

This option appears on the **Certificates** page of the **Operation Settings** workflow.

Set this option to **Yes** to ensure that any derived credentials created do not exceed the lifetime of the deriving certificate. If the lifetime of the derived credential (as determined by the **Lifetime** setting in the credential profile or the `MaxRequestExpiryDate` set for the person in the Lifecycle API) is greater than the lifetime of the presented certificate, the lifetime of the derived credential is lowered to match the expiry date of the deriving certificate.

The default is **No**.

Note: Some CAs do not allow control over the time portion of the certificate expiry. When MyID sets the lifetime of the derived credential, the date is aligned with the lifetime of the deriving certificate, but the time may not match exactly, depending on the certificate authority being used.

- **App Download URL – ANDROID and App Download URL – iOS**

These options appear on the **Issuance Processes** tab of the **Operation Settings** workflow. The URLs are embedded in the QR code displayed to the user that allows them to download the appropriate version of the mobile app.

It is important that if the hosting MyID system has any kind of LDAP sync enabled, such as background update, that the **Synchronize new accounts with directory** configuration option is configured **ON**. Failing to do this may cause inconsistent behavior due to LDAP synchronization schedules.

Note: Group default roles relate only to the **Add Person** and **Edit Person** workflows, and as such are not applied to users imported through SSRP. Roles that are configured to be imported from LDAP will be assigned to the newly-created user account. Any roles applied to user accounts by SSRP override any role restrictions in MyID.

2.4.1 Setting the credential check period

By default, seven days after MyID issues derived credentials, it checks the original credentials that were used to request the derived credentials. If, during this period, the original credentials became no longer valid (for example, if the smart card was canceled), MyID revokes the derived credentials.

The full device is canceled, not individual certificates on the device. If the device has archived certificates issued as derived credentials, these are also revoked, in addition to the authentication and signing certificates.

Note: MyID does not distinguish between the certificate being suspended or revoked; if it is on the CRL, it revokes the derived credentials.

The reason for cancellation is included in the audit information for troubleshooting purposes; this states that it was due to the PIV certificate being revoked. If your system is configured for device cancellation notifications, these are sent for the revoked derived credentials.

You must make sure that MyID can access the CRL. If the CRL is not available, MyID does not carry out any revocation and logs the error in the audit trail. There may be a lag between the PIV issuer revoking the PIV credential and the CRL being updated and republished.

You must make sure that the PIV Issuer carries out PIV card revocation in appropriate situations; this feature relies on this step occurring to identify and trigger the revocation of derived credentials.

You can adjust the time period for the credential check.

Alternatively, you can configure MyID to repeat the revocation check at regular intervals. In this case, MyID checks the status of the original credentials at the specified interval until the issued derived credentials are canceled or have expired.

To configure the credential checks:

1. From the **Configuration** category, select **Operation Settings**.
2. On the **Certificates** tab, set the following:
 - **Derived credential revocation check offset** – set to the number of days after issuing derived credentials that you want MyID to check the original credentials.
 - **Derived Credential Revocation Check Interval** – set to the number of hours between repeated checks of the original credentials. By default this is 0, which means that the check is not repeated.

Note: If you set this option to a value greater than 0, it overrides the **Derived credential revocation check offset** setting.

3. Click **Save changes**.

2.4.2 Determining which cards are available for derived credentials

You may want to configure your system to issue derived credentials only from cards that have been issued by specific federal agencies. To do this, you can match the agency code in the FASC-N.

To determine which cards you can use to request derived credentials:

1. From the **Configuration** category, select the **Operation Settings** workflow.
2. Click the **Certificates** tab.
3. Set the following options:

- **Cards Allowed For Derivation**

Set this option to a regular expression that will be matched against the ASCII version of the card's FASC-N to determine whether the card can be used to request derived credential. If the regular expression matches, the card can be used.

For example:

```
5400.+
```

This example allows any card from the agency with code 5400 to be used. The agency code appears at the start of the ASCII FASC-N.

```
((5400)|(7280)).+
```

This example allows any card from the agency with code 5400 or the agency with code 7280 to be used.

Note: By default, this option is blank, which means that no cards can be used to request derived credentials. To allow all cards to be used, use the following regular expression:

```
.+
```

4. Click **Save changes**.

2.4.3 Configuring certificate OIDs checked on PIV cards

When a PIV card is presented to the SSRP, MyID verifies that the cardholder can perform two factor authentication with the PIV card, performing the PKI-AUTH check to verify the PIV-Authentication certificate.

Additionally, MyID verifies the Digital Signature certificate.

These certificate checks ensure that the certificate is valid and was issued from a CA that chains up to a root certificate in the `DerivedCredentialTrustedRoots` store.

It also checks that the end-user certificate contains the correct OID to mark it as a PIV-Authentication or Digital Signature certificate.

By default, MyID is configured with the OIDs required by FIPS201-2; however, you can change the OIDs if required (for example, for a CIV certificate).

To configure the OIDs:

1. From the **Configuration** category, select **Operation Settings**.
2. On the **Certificates** tab, set the following:
 - **Derived credential certificate OID** – set this to the OID to be checked on the PIV Authentication certificate.
The default value is
`2.16.840.1.101.3.2.1.3.13`
 - **Derived credential signing certificate OID** – set this to the a semicolon-delimited list of OIDs to be checked on the Digital Signature certificate.
The default value is
`2.16.840.1.101.3.2.1.3.6;2.16.840.1.101.3.2.1.3.7;`
`2.16.840.1.101.3.2.1.3.16`
3. Click **Save changes**.

2.5 Configuring email notifications

You can configure MyID to send a notification to the credential owner when a derived credential is requested. This email message contains information on the owner, the certificate, and the job that was created for the request of the derived credential.

2.5.1 Setting up email

To set up MyID to enable email notifications, see the *Setting up email* section in the [Advanced Configuration Guide](#).

2.5.2 Editing the request email template

To edit the email template:

1. From the **Configuration** category, select **Email Templates**.

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. Select the **Derived Credential Requested** template, then click **Modify**.

The screenshot shows the 'Edit Email Template' window for the 'Derived Credential Requested' template. The 'Subject' field is 'Derived Credential Requested'. The 'Template Name' is 'Derived Credential Requested'. The 'Template Description' is 'Sent to a user who has been requested a derived credential'. The 'Enabled' checkbox is checked. The 'Template Body' contains HTML code for an email template. The 'Standard substitutions' list includes %n (New Line), %t (Tab), %x (Webserver URL Path), and %jobid (Job ID). The 'Substitution Legend' defines codes like %dn (Distinguished name), %sn (Certificate serial number), %expiry (Certificate expiry), %issuer (Issuer name), %Person:vPeopleUserAccounts:LogonName (Logon name), %Job:vJobsWithJobID:JobID (Job ID), %Job:vJobsWithJobID:Status (Status), %Job:vJobsWithJobID:InitiationDate (Initiation date), and %Job:vNewRequestEmailCodes:CredentialProfileName (Credential profile). The 'Transport' is set to 'email' and 'Signed' is checked. 'Save' and 'Cancel' buttons are at the bottom right.

3. Select the **Enabled** option to enable or disable the template.

Disabling the template prevents the notifications from being sent.

4. Edit the **Template Body**.

The body contains HTML text, and allows you to include codes in the template that are substituted for information about the request when the email is sent.

You can use the following substitution codes:

- %dn – Distinguished name.
- %sn – Certificate serial number.
- %expiry – Certificate expiry date.
- %issuer – Issuer name
- %Person:vPeopleUserAccounts:LogonName – Logon name of the credential owner.
- %Job:vJobsWithJobID:JobID – ID of the request job.
- %Job:vJobsWithJobID:Status – status of the request job.
- %Job:vJobsWithJobID:InitiationDate – initiation date of the request job.
- %Job:vNewRequestEmailCodes:CredentialProfileName – credential profile requested for the derived credential.

5. Click **Save**.

2.5.3 Editing the cancellation email template

To edit the email template:

1. From the **Configuration** category, select **Email Templates**.

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. Select the **Cancel Card** template, then click **Modify**.

The screenshot shows the 'Edit Email Template' interface for the 'Cancel Card' template. The breadcrumb trail at the top is 'Email Templates > Select Email Template > Edit Email Template'. The 'Edit Email Template' tab is active. The form contains the following fields:

- Subject:** Cancel Card
- Template Name:** Cancel Card
- Template Description:** Sent to a user when one of their devices is cancelled
- Enabled:** ☒
- Template Body:** A text area containing HTML code:

```
<table>
  <tr>
    <td>Your device
    serial number %Device:vDevicesWithDeviceID:SerialNumber,
    type %Device:vDevicesWithDeviceID:DeviceTypeName has been
    cancelled.</td>
  </tr>
  <tr>
    <td>If you did not
    expect to receive this email, please contact your
    administrator.</td>
  </tr>
</table>
```
- Standard substitutions:** A list of substitution codes: %n - New Line, %t - Tab, %x - Webserver URL Path, %jobid - Job ID.
- Edit substitution:** A link to edit the substitutions.
- Substitution Legend:** None defined
- Transport:** email (selected from a dropdown)
- Signed:** ☒

At the bottom right, there are 'Save' and 'Cancel' buttons.

3. Select the **Enabled** option to enable or disable the template.

Disabling the template prevents the notifications from being sent.

4. Edit the **Template Body**.

The body contains HTML text, and allows you to include codes in the template that are substituted for information about the request when the email is sent.

You can use the following substitution codes:

- %Device:vDevicesWithDeviceID:SerialNumber – serial number of the canceled device.
- %Device:vDevicesWithDeviceID:DeviceTypeName – type of the canceled device.

5. Click **Save**.

2.5.4 Obtaining the email address

If the user is unknown to MyID (that is, the original credential was issued by a different system):

- the **Synchronize new accounts with directory** configuration option is set, if MyID can link the account to the directory and that directory account contains an email address, it sends the notification to that address; otherwise, it sends the email notification to the address from the deriving certificate.
- If the **Synchronize new accounts with directory** configuration option is *not* set, MyID attempts to obtain the email address from the deriving certificate and sends the notification to that address.

If the user is known to MyID (that is, there is already an account in MyID for the user):

- If the **Update email address from derivation** configuration option is set, MyID attempts to obtain the email address from the deriving certificate and sends the notification to that address; it also updates the email address in the MyID account with the address from the certificate. If there is no email address in the certificate, it uses the email address in the MyID account.
- If the **Update email address from derivation** configuration option is *not* set, MyID sends the notification to the email address in the MyID account.

If MyID cannot obtain an email address from any source, it does not attempt to send an email notification.

2.5.5 Requiring an email address

You are recommended to configure the credential profile for the derived credential to require an email address.

To require an email address:

1. From the **Configuration** category, select **Operation Settings**.

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

2. Click the Issuance Processes page.

3. Set the following option:

- **Requisite User Data** – set this option to Yes.

This option makes the Requisite User Data section appear in the Credential Profiles workflow.

4. Click **Save changes**.

5. From the **Configuration** category, select **Credential Profiles**.

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

6. Click **New** to create a new credential profile, or select an existing credential profile and select **Modify**.
7. Select the **Requisite User Data** option.

Credential Profiles > Select Profile > Credential Profile Details > Select Certificates > Select Applets > Select Roles > Select Card Layout > Add Comments

Credential Profile

Name: **Derived Credential** Description:

Device Friendly Name:

Requisite User Data

	Not Required for Request	Required for Request	Required for Validate / Collect	Allowed Value(s) (Optional)
Address 1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
Address 2	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
Cell	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
City	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
Email	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>
Employee ID	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
PIV Distinguished Name	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
State + Zip	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
User Principal Name	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
User SID	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>

Next

8. Set the **Email** option to **Required for Request**.

9. Complete the rest of the credential profile configuration.

2.5.6 Updating the email address

If you set the **Update email address from derivation** option (on the **Certificates** page of the **Operation Settings** workflow) to Yes, if MyID obtains an email address from the deriving certificate, it updates the person's record within MyID with this address.

2.6 Granting access to the workflows

The system makes use of the following workflows:

- **Cancel Credential** – used within MyID to cancel a mobile ID and revoke its certificates.
- **Enable / Disable ID** – used within MyID to enable or disable a mobile ID, and suspend or enable its certificates.
- **Unlock Credential** – used within MyID to retrieve an unlock code for an issued mobile ID.
- **Collect My Updates** – used by the Identity Agent app to update mobile IDs after issuance.
- **Issue Device** – used by the Identity Agent app to obtain a mobile ID.
- **Collect My Card** – used in the Self-Service App to collect VSCs.

Use the **Edit Roles** workflow to grant access for these workflows to the roles you want to be able to access them.

2.6.1 Role permissions

You must use the **Edit Roles** workflow to ensure that the roles used for derived credentials have the appropriate permissions.

The roles must have access to the following:

- **Collect My Updates**
- **Issue Device**
- **Collect My Card**

2.7 Setting up the certificate checks

When working with client certificate-based derived credentials, for the derived credential certificate checks to work, you must export the certificate authority's root certificate, then install this on your MyID application server.

This is not required for OpenID-based derived credentials.

Note: The RootCA certificate (the certificate authority's root certificate) must be trusted by the MyID application server. If it is not already a trusted certificate, add it to the Trusted Root Certificate Authority store for the local machine.

2.8 Setting up the credential profiles for derived credentials

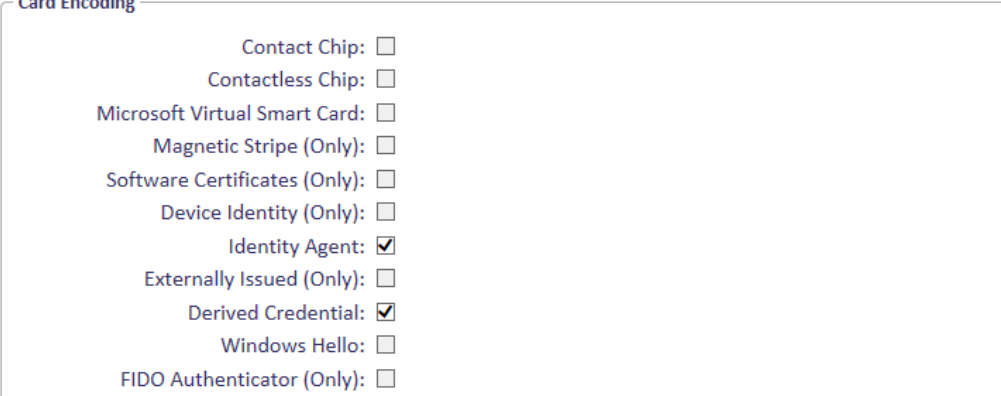
You must create new credential profiles for the derived credentials.

You must create at least one credential profile to contain the certificates that you want to issue to the derived credential. You may create as many of these credential profiles as you need; for example, you may want to create a credential profile for mobile devices and a credential profile for Microsoft VSCs.

2.8.1 Creating an Identity Agent credential profile

To create a credential profile for issuing derived credentials to mobile devices:

1. From the **Configuration** category, select **Credential Profiles**.
2. Click **New**.
3. Type a **Name** for the credential profile.
4. In **Card Encoding**, select **Identity Agent** and **Derived Credential**.



Card Encoding

- Contact Chip: ☐
- Contactless Chip: ☐
- Microsoft Virtual Smart Card: ☐
- Magnetic Stripe (Only): ☐
- Software Certificates (Only): ☐
- Device Identity (Only): ☐
- Identity Agent: ☒
- Externally Issued (Only): ☐
- Derived Credential: ☒
- Windows Hello: ☐
- FIDO Authenticator (Only): ☐

5. In **Services**, make sure **MyID Logon** and **MyID Encryption** are selected.
Note: If you select the **Identity Agent** option *after* you select the **Derived Credential** option, you cannot select the **Services** option; however, **MyID Logon** and **MyID Encryption** are automatically selected.
6. In **Issuance Settings**, in the **Mobile Device Restrictions** drop-down list, select one of the following:
 - **Any** – The mobile identity can be loaded onto any mobile.
 - **Known Mobiles** – The mobile identity can be loaded onto any mobile that has already been registered with MyID. See the *Setting up the Identity Agent credential profiles* section in the [Mobile Identity Management](#) guide for details.
 - **My Mobiles Only** – The mobile identity can be loaded only onto mobiles associated with the user's account.
7. If you are issuing Identity Agent credentials for users associated with cards that were not issued by the current system, set the following option:
 - **Require Facial Biometrics** – Never Required.
8. For mobile derived credentials issued through an MDM, if you want to issue the credential to a device that is already issued to the target user, set the following option:
 - **Issue over Existing Credential** – set this option, and if the device is already issued to the target user, it is automatically canceled and then the new device issued. Existing signing certificates are revoked, but existing archived certificates are not revoked. If the device is issued to a different user, the collection fails.
Note: The credential profile used for the existing issuance does not affect this behavior; existing credentials are overwritten only if the credential profile for the new credential has the **Issue over Existing Credential** option set.

9. In **Device Profiles**, select the appropriate data model file from the **Card Format** drop-down list.

See the *Setting up the Identity Agent credential profiles* section in the [Mobile Identity Management](#) guide for details.

10. If you want to display collection instructions specific to this credential profile to the user through the Self-Service Request Portal when the user is collecting their mobile identity document, in **Collection Instructions**, type plain text instructions into the box. If you leave this blank, the translatable text for collection instructions is shown instead.

The translatable text for collection instructions has the identifying string `ID1009`; see section 6.1, *Customizing the terminology* for more information.

If you want to access the Self-Service Request Portal from a mobile device, you can set the following options:

- **Show app collection link on mobile OS** – determines which app to launch when you request a mobile derived credential from the Self-Service Request Portal on a mobile device (iOS or Android). This allows you to collect the mobile derived credential on the same mobile device that you are using to request the derived credential. You can select one of the following options:
 - **None** – do not display an app collection link.
 - **MyID Mobile Identity Document Wallet** – display a link for the MyID Wallet app.
 - **BlackBerry UEM Client** – display a link for the BlackBerry UEM Client app.

Note: If your organization requires an app that is not listed here, contact customer support quoting reference SUP-393 to discuss your requirements.

- **Allow QR Code on mobile OS** – if you access the Self-Service Request Portal on a mobile device, and you have selected anything other than **None** for the **Show app collection link on mobile OS** option, SSRP does not display a QR code. If you want to display this QR code (for example, if you want to allow users to request the derived credential on one mobile device and collect on another) select this option; SSRP displays both a collection link and the QR code, allowing the end user to choose which method to use to collect their credentials.

If you select **None** for the **Show app collection link on mobile OS** option, you do not need to select this option; the Self-Service Request Portal displays a QR code.

These options take effect when you access the Self-Service Request Portal from a browser that reports its operating system as iOS or Android.

11. Click **Next**.
12. Select the certificates you want to make available.

All of the certificates you select here will be issued to your mobile device.

You can select the archived and historic certificate options on this screen. See the *Import and distribute certificates to devices* section in the [Administration Guide](#) for details of the **Issue new**, **Use existing**, and **Historic Only** options.

13. Click **Next** and proceed to the Select Roles screen.

14. Select the roles you want to be able to issue this credential profile, and the roles you want to be able to be issued this credential profile.
Note: Any role to which you want to issue derived credentials must have the **Issue Device** option selected in the **Cards** category within the **Edit Roles** workflow.
15. Click **Next**.
16. Select the card layouts you want to make available to the mobile device.
Badges based on these layouts will be transferred to the mobile device as part of the mobile ID. Note, however, that the reverse sides of the selected layouts (the `_back` layouts) will not be available on the mobile device.
Note: You must select at least one card layout. If you do not want to display personalized badge information on the mobile device, create a card layout containing default artwork and no user information.
17. Select one of the layouts to be the default layout.
This layout will be displayed by default when using the Identity Agent app, and will be used for phone-to-phone identity verification.
18. Click **Next**.
19. Type your **Comments**, and complete the workflow.

2.8.2 Creating a mobile identity document credential profile

To create a credential profile for issuing derived mobile identity documents to mobile devices:

1. From the **Configuration** category, select **Credential profiles**.
2. Click **New**.
3. Type a **Name** for the credential profile.
4. For the **Card Encoding**, select **Mobile Identity Document** and **Derived Credential**.
5. In **Issuance Settings**, in the **Mobile Device Restrictions** drop-down list, select one of the following:
 - **Any** – The mobile identity document can be loaded onto any mobile.
 - **Known Mobiles** – The mobile identity document can be loaded onto any mobile that has already been registered with MyID. See the *Setting up the Identity Agent credential profiles* section in the [Mobile Identity Management](#) document for details.
 - **My Mobiles Only** – The mobile identity document can be loaded only onto mobiles associated with the user's account.

You may also set the other issuance settings available to standard mobile identity document credential profiles. See the *Creating the mobile identity document credential profile* section in the [Mobile Identity Documents](#) guide for details.

6. If you want to display collection instructions specific to this credential profile to the user through the Self-Service Request Portal when the user is collecting their mobile identity document, in **Collection Instructions**, type plain text instructions into the box. If you leave this blank, the default translatable text for collection instructions is shown instead.

The translatable text for collection instructions has the identifying string `ID1009`; see section 6.1, [Customizing the terminology](#) for more information.

If you want to access the Self-Service Request Portal from a mobile device, you can set the following options:

- **Show app collection link on mobile OS** – determines which app to launch when you request a mobile derived credential from the Self-Service Request Portal on a mobile device (iOS or Android). This allows you to collect the mobile derived credential on the same mobile device that you are using to request the derived credential. You can select one of the following options:
 - **None** – do not display an app collection link.
 - **MyID Mobile Identity Document Wallet** – display a link for the MyID Wallet app.
 - **BlackBerry UEM Client** – display a link for the BlackBerry UEM Client app.

Note: If your organization requires an app that is not listed here, contact customer support quoting reference SUP-393 to discuss your requirements.

- **Allow QR Code on mobile OS** – if you access the Self-Service Request Portal on a mobile device, and you have selected anything other than **None** for the **Show app collection link on mobile OS** option, SSRP does not display a QR code. If you want to display this QR code (for example, if you want to allow users to request the derived credential on one mobile device and collect on another) select this option; SSRP displays both a collection link and the QR code, allowing the end user to choose which method to use to collect their credentials.

If you select **None** for the **Show app collection link on mobile OS** option, you do not need to select this option; the Self-Service Request Portal displays a QR code.

These options take effect when you access the Self-Service Request Portal from a browser that reports its operating system as iOS or Android.

7. Click **Next**.

8. Select the roles you want to be able to issue this credential profile, and the roles you want to be able to be issued this credential profile.

Note: Any role to which you want to issue derived credentials must have the following configured in the **Edit Roles** workflow:

- Select the **Issue Device** option in the list of workflows.
- Select the **Collect My Card** option in the list of workflows.
- Select the **Password** option in the **Logon Methods**.

9. Click **Next**.

10. Select one of the layouts to be the default layout.

This layout will be used by your identity app, and will be used for phone-to-phone identity verification.

11. Click **Next**.

12. Type your **Comments**, and complete the workflow.

2.8.3 Creating a VSC credential profile

To create a credential profile for issuing derived credentials as Microsoft VSCs:

1. From the **Configuration** category, select **Credential profiles**.
2. Click **New**.
3. Type a **Name** for the credential profile.
4. For the **Card Encoding**, select **Microsoft Virtual Smart Card** and **Derived Credential**.
5. In **Services**, make sure **MyID Logon** and **MyID Encryption** are selected.
6. In **Issuance Settings**, set the following options:

- **Generate Code on Request** – select one of the following:
 - **None** – no logon code is generated.
 - **Simple Logon Code** – the logon code is generated using the complexity rules as defined by the **Simple Logon Code Complexity** configuration option.
 - **Complex Logon Code** – the logon code is generated using the complexity rules as defined by the **Complex Logon Code Complexity** configuration option.

Note: To be FIPS 201-3 compliant, you must select **Simple** or **Complex**. See the *Logon using security phrases* section in the [Administration Guide](#) for details of configuring the logon code complexity.

Important: You must set the **Allow Logon Codes** option (on the **Logon** page of the **Security Settings** workflow) to Yes to allow MyID to use logon codes.

- **Credential Group** – if you want to restrict users to have a single derived credential VSC, type an identifier here; for example:

DC VSC

If you set the **Active credential profiles per person** configuration option (on the **Issuance Processes** page of the **Operation Settings** workflow) to **One per credential group**, MyID ensures that the user can have only one credential with the same **Credential Group** name.

- **Cancel Previously Issued Device**

This option works in conjunction with the **Credential Group** setting. Select this option, and MyID cancels any previously-issued credentials instead of disabling them. When you collect the new VSC using the Self-Service App (and you have the **Erase Unused VSCs** permission for your role, as configured in the **Edit Roles** workflow) the Self-Service App will delete any of the canceled VSCs on your device.

For more information on these options, see the *Additional credential profile options* section in the [Administration Guide](#).

7. For Microsoft VSCs, set the PIN to 16 numeric digits if you want to ensure that the derived credential is compliant with FIPS 201-3.
 - a. In **PIN Settings**, set the **Maximum PIN Length** and **Minimum PIN Length** options to 16.
 - b. In **PIN Characters**, set **Numeric** to **Mandatory**, and **Lowercase**, **Uppercase**, and **Symbol** to **Not Allowed**.

8. Click **Next**.
9. Select the certificates you want to make available.
All of the certificates you select here will be issued to your VSC.
You can select the archived and historic certificate options on this screen. See the *Selecting certificates* section in the [Administration Guide](#) for details of the **Issue new**, **Use existing**, and **Historic Only** options.
10. Click **Next** and proceed to the Select Roles screen.
11. Select the roles you want to be able to issue this credential profile, and the roles you want to be able to be issued this credential profile.
Note: Any role to which you want to issue derived credentials must have the following configured in the **Edit Roles** workflow:
 - Select the **Issue Device** option in the list of workflows.
 - Select the **Collect My Card** option in the list of workflows.
 - Select the **Password** option in the **Logon Methods**.
12. Click **Next**.
13. Click **Next**.
14. Type your **Comments**, and complete the workflow.

2.8.4 Creating a Windows Hello credential profile

Important: The **Windows Hello** option in the credential profile appears only when you have set the **Windows Hello for Business supported within MyID** configuration option. See the *Setting the Windows Hello configuration options* section in the [Windows Hello for Business Integration Guide](#) for details.

To create a credential profile for issuing derived credentials to Windows Hello:

1. From the **Configuration** category, select **Credential Profiles**.
2. Click **New**.
3. Type a **Name** and **Description**.
4. In the **Card Encoding** section, select **Windows Hello** and **Derived Credential**.
5. In the **Services** section, select **MyID Logon** and **MyID Encryption**.
6. In **Issuance Settings**, set the following options:
 - **Generate Code on Request** – select one of the following:
 - **None** – no logon code is generated.
 - **Simple Logon Code** – the logon code is generated using the complexity rules as defined by the **Simple Logon Code Complexity** configuration option.
 - **Complex Logon Code** – the logon code is generated using the complexity rules as defined by the **Complex Logon Code Complexity** configuration option.

Note: To be FIPS 201-3 compliant, you must select **Simple** or **Complex**. See the *Logon using security phrases* section in the [Administration Guide](#) for details of configuring the logon code complexity.

Important: You must set the **Allow Logon Codes** option (on the **Logon** page of the **Security Settings** workflow) to Yes to allow MyID to use logon codes.

7. In the **Mail Documents** section, set up any mailing documents you may want to issue.

See the *Mail Documents* section in the [Administration Guide](#) for details.

8. Click **Next**.

9. On the Select Certificates screen, select the certificates you want to issue to the Windows Hello credential.

Note: You must use a certificate for **Signing** and **Encryption**; you cannot use MyID keys for signing and encryption operations on Windows Hello credentials.

For more information on this screen, see the *Selecting certificates* section in the [Administration Guide](#).

10. Click **Next** and proceed to the Select Roles screen.

Note: Any role to which you want to issue derived credentials must have the following configured in the **Edit Roles** workflow:

- Select the **Issue Device** option in the list of workflows.
- Select the **Collect My Card** option in the list of workflows.
- Select the **Password** option in the **Logon Methods**.

See the *Linking credential profiles to roles* section in the [Administration Guide](#) for details.

11. Click **Next** and complete the workflow.

You do not need to specify any card layouts.

2.8.5 Creating a FIDO authenticator credential profile

See the [Passkey Integration Guide](#) for details of setting up MyID to issue FIDO authenticators.

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.

Credential Profile

Name:

Description:

Device Friendly Name:

Card Encoding

- Card Encoding
- Services
- Issuance Settings
- Self-Service Unlock Authentication
- PIN Settings
- PIN Characters
- Biometric Settings
- Mail Documents
- Credential Stock
- Device Profiles
- Authentication Types
- FIDO Settings
- Requisite User Data

Card Encoding

- Contact Chip: ☐
- Contactless Chip: ☐
- Microsoft Virtual Smart Card: ☐
- Magnetic Stripe (Only): ☐
- Software Certificates (Only): ☐
- Device Identity (Only): ☐
- Identity Agent: ☐
- Externally Issued (Only): ☐
- Derived Credential: ☒
- Windows Hello: ☐
- FIDO Authenticator (Only): ☒

Next

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[BGI1OQDSZ], 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:

Credential Profile

Name:

Description:

Device Friendly Name:

FIDO Settings

- Assurance Level: **High**
- User Verification: **Required**
- Authenticator Type: **Removable (e.g. USB or sr)**
- Require Client Side Discoverable Key: ☐
- Require Attestation: **Basic**
- Immediate registration via Self-Service Request Portal: ☐
- Authentication Server: **MyID CMS**
- Automatically Revoke at Expiry: ☐

Next

- **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**.

- **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.
- **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`).
 - **Enterprise** – carry out an enterprise attestation check during the registration process.
 - **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 **Mobile** 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.

2.8.6 Creating a credential profile for other devices

To create a credential profile for issuing derived credentials to any other type of device (for example, smart cards and USB tokens):

1. From the **Configuration** category, select **Credential profiles**.
2. Click **New**.
3. Type a **Name** for the credential profile.
4. For the **Card Encoding**, select **Contact Chip** and **Derived Credential**.
5. In **Services**, make sure **MyID Logon** and **MyID Encryption** are selected.
6. In **Issuance Settings**, set the following option:
 - **Generate Code on Request** – select one of the following:
 - **None** – no logon code is generated.
 - **Simple Logon Code** – the logon code is generated using the complexity rules as defined by the **Simple Logon Code Complexity** configuration option.
 - **Complex Logon Code** – the logon code is generated using the complexity rules as defined by the **Complex Logon Code Complexity** configuration option.

Note: To be FIPS 201-3 compliant, you must select **Simple** or **Complex**. See the *Logon using security phrases* section in the [Administration Guide](#) for details of configuring the logon code complexity.

Important: You must set the **Allow Logon Codes** option (on the **Logon** page of the **Security Settings** workflow) to Yes to allow MyID to use logon codes.

7. In **Device Profiles**, if the devices to which you want to issue the derived credentials require a card format file (for example, to use a PIV data model), select the appropriate file from the **Card Format** drop-down list.

See the [Smart Card Integration Guide](#) for information on the card format files required for your devices.

8. Click **Next**.

9. Select the certificates you want to make available.

- For credential profiles that use a PIV data model, select the PIV containers for the certificates. To allow online unlocking, you must include a certificate in the PIV Card Authentication Certificate container.
- For credential profiles that do not use a PIV data model, do not select any containers.

All of the certificates you select here will be issued to your device.

You can select the archived and historic certificate options on this screen. See the *Selecting certificates* section in the [Administration Guide](#) for details of the **Issue new**, **Use existing**, and **Historic Only** options.

10. Click **Next** and proceed to the Select Roles screen.

11. Select the roles you want to be able to issue this credential profile, and the roles you want to be able to be issued this credential profile.

Note: Any role to which you want to issue derived credentials must have the following configured in the **Edit Roles** workflow:

- Select the **Issue Device** option in the list of workflows.
- Select the **Collect My Card** option in the list of workflows.
- Select the **Password** option in the **Logon Methods**.

12. Click **Next**.

13. Click **Next**.

14. Type your **Comments**, and complete the workflow.

2.8.7 Credential profile restrictions

Note: At the point of the request for the derived credential, full details about the user are not known; this means that MyID cannot verify some credential profile requirements, including the requirement for facial and fingerprint biometrics, as well as the enforcement of a UPN or email address. You are recommended *not* to apply these restrictions to a credential profile used for derived credentials as, if these values are not available, the user will be unable to collect the derived credential.

If an existing user account is found, the available credential profiles are restricted to profiles that are available to the roles that are specified by the list of roles held in MyID by the existing user, *if* they are also included in the roles specified in the `ssrp.conf.xml` file.

If a user was not found in MyID, and SSRP is configured to import unknown users, the credential profile selection is based on the roles from `ssrp.conf.xml` only.

2.8.8 Configuring the available credential profiles

You can edit the `ssrp.conf.xml` configuration file on the MyID application server to configure which credential profiles are available through the SSRP.

2.8.9 Mapping certificates to roles and credential profiles

For client certificate-based derived credentials, you can configure the system to make specific credential profiles available to users based on the user certificates on their original smart cards. To do this, you set up a mapping between the OIDs of the possible certificates and the roles you have set up within MyID; if the user has a certificate that matches the listed OIDs, they are given the specified roles, and therefore granted access to any credential profiles for derived credentials that are available to these roles.

2.8.9.1 Example

You have configured three roles:

- **Derived Credential User**
- **Secure Access**
- **Remote Access**

You have configured four credential profiles for derived credentials:

- **Standard DC mobile** – available to the **Derived Credential User** role.
- **Standard DC VSC** – available to the **Derived Credential User** role.
- **Secure access mobile** – available to the **Secure Access** role.
- **Remote access VSC** – available to the **Remote Access** role.

You set up the mappings as follows:

- **Derived Credential User:**
 - Any OID.
- **Secure Access:**
 - 1.2.826.0.1.2697033.1.1
- **Remote Access:**
 - 2.16.840.1.101.3.2.1.6.1
 - 2.16.840.1.101.3.2.1.6.2
 - 2.16.840.1.101.3.2.1.6.3
 - 2.16.840.1.101.3.2.1.6.4

If a user presents a credential with no matching OIDs, they are allocated the **Derived Credential User** role, and therefore can choose one of the following credential profiles:

- **Standard DC mobile.**
- **Standard DC VSC.**

If a user presents a credential with the following matching OIDs:

- 1.2.826.0.1.2697033.1.1
- 2.16.840.1.101.3.2.1.6.1

- 2.16.840.1.101.3.2.1.6.2
- 2.16.840.1.101.3.2.1.6.3
- 2.16.840.1.101.3.2.1.6.4

they are allocated the **Derived Credential User** role, the **Secure Access** role, and the **Remote Access** role, and therefore can choose any of the following credential profiles:

- **Standard DC mobile.**
- **Standard DC VSC.**
- **Secure access mobile.**
- **Remote access VSC.**

If a user presents a credential with the following matching OIDs:

- 1.2.826.0.1.2697033.1.1
- 2.16.840.1.101.3.2.1.6.1

they are allocated the **Derived Credential User** role and the **Secure Access** role, but not the **Remote Access** role – they match *some*, but not *all* of the OIDs required for remote access. Therefore they can choose from the following credential profiles:

- **Standard DC mobile.**
- **Standard DC VSC.**
- **Secure access mobile.**

2.8.10 Restricting based on the certificate authority path

For client certificate-based derived credentials, you can further restrict the available role based on the path of the CA that issued the certificate used to make the request – you can specify a DN that must be included in the SSL certificate's chain to be eligible. If the DN is not present, the role is not allowed.

2.8.11 Verifying certificates

For client certificate-based derived credentials, you can configure the system to perform a real-time certificate validity check before requesting the derived credential. If the check fails, the issuance is prevented – even if the user selects a credential profile from a different role.

Certificate validation occurs using the Microsoft WinCrypt API.

2.8.12 Configuration file format

For client certificate-based derived credentials, the `ssrp.conf.xml` configuration file is stored on the MyID application server in the following location:

```
C:\Program Files\Intercede\MyID\Settings\
```

Within the top-level `<roles>` node, you can add one or more `<role>` nodes.

Within this `<role>` node, you can add the following nodes:

- `<OID>` (optional) – specify an OID that must be present on the user certificate. You can include multiple `<OID>` nodes; the certificate must match *all* specified OIDs.
- `<CAPath>` (optional) – specify a DN that must be included in the SSL certificate's chain.

- `<VerifyCertificate>` (optional) – set to `true` to perform a real time certificate validity check before requesting the derived credential.
- `<ImportPIVDN>` (optional) – set to `true` if you want to populate the PIV DN (`Xu55`) field from the deriving credential. If you set this to `false`, or if the option is missing, this field is left blank.

Note: By default, from MyID 12.5 onwards, the value is set to `true` for the default role of 984, while in previous releases the node was absent and so defaulted to `false`.

- `<person>` (optional) – provide a list of additional attributes and values from the `vPeopleUserAccounts` view that must be matched in addition to the UPN and DN matching. You can use this to disambiguate between accounts that share the same DN. Provide the attributes and values in subnodes; for example if you have an attribute `XuPrimary` that you use to mark the primary accounts, you could use:

```
<person>
  <XuPrimary>Y</XuPrimary>
</person>
```

You can include multiple attributes if necessary.

- `<role>` – specify the role that will be granted. Within this node, you must include the following parameters:
 - `userprofileid` – the ID from the `UserProfiles` table in the MyID database for the role.
 - `UserProfileName` – the Name from the `UserProfiles` table in the MyID database for the role.
 - `scope` – set to 1 (self) for derived credential users. Other scope values are for operators and administrators.
 - `logonmechanism` – set to 1 for password logon, 2 for smart card logon. If you want to allow multiple methods of logging in, repeat the same `<role>` node and supply a different `logonmechanism` value.

Optionally, you can add the following attribute:

- `doNotUpdateUser` – set to `true`, and if an existing user is found within MyID, the role is not added to the user's list of roles, but is used only to filter the available credential profiles. If this setting is not present, or set to `false`, the role is added to the existing user's list of roles as part of the SSRP enrollment.

Example:

```
<?xml version="1.0" encoding="utf-8" ?>
<roles>
  <role>
    <role userprofileid="984" UserProfileName="Derived Credential
User" scope="1" logonmechanism="0" />
    <role userprofileid="984" UserProfileName="Derived Credential
User" scope="1" logonmechanism="1" />
    <ImportPIVDN>true</ImportPIVDN>
  </role>
</role>
```

```

    <OID>1.2.826.0.1.2697033.1.1</OID>
    <role userprofileid="21" UserProfileName="Secure Access" scope="1" logonmechanism="1"
  />
</role>
<role>
  <OID>2.16.840.1.101.3.2.1.6.1</OID>
  <OID>2.16.840.1.101.3.2.1.6.2</OID>
  <OID>2.16.840.1.101.3.2.1.6.3</OID>
  <OID>2.16.840.1.101.3.2.1.6.4</OID>
  <CAPath>dc=VPN,o=intercede,o=com</CAPath>
  <VerifyCertificate>true</VerifyCertificate>
  <role userprofileid="20" UserProfileName="Remote
Access" scope="1" logonmechanism="1" doNotUpdateUser="true"/>
</role>
<role>
  <person>
    <Xu4>4</Xu4>
    <Xu41>AP</Xu41>
  </person>
  <role userprofileid="984" scope="1" />
  <ImportPIDN>true</ImportPIDN>
</role>
</roles>

```

2.9 Lifecycle management

Subsequent lifecycle management of your issued derived credentials may rely on notifications that are sent to email addresses; in this case, you must add this information to the user account in MyID (using, for example, directory synchronization or the Lifecycle API) after the request for the derived credential has been made.

Additionally, if the MyID logon name is not known to the client, you must ensure that the logon name is passed through the link in the email notification.

To do this, edit the following email templates:

- Renew Certificate Notification First
- Renew Certificate Notification Second
- Renew Certificate Notification
- Issue Card Notification
- Apply Update Notification
- Replacement Card Notification

Using the **Email Templates** workflow, edit each template, and change the link that opens the Self-Service App from:

```
myidssa:///w+/jobid:%j
```

to:

```
myidssa:///w+/jobid:%j+un:{%logonName:URI}
```

See the *Changing email messages* section in the [Administration Guide](#) for details of using the **Email Templates** workflow.

2.10 Request validation

It is not intended for a derived credential to require secondary validation. However, if this *is* configured, it is essential that the user's email address is obtained and recorded before performing the validation. Without the email address, it will not be possible to inform the user that their request has been approved and provide them a link to collect their derived credential.

The one exception to this is the situation where a VSC is requested, the target machine is domain joined, and the MyID Self-Service App is installed and configured to check for jobs at logon. When this process next checks for jobs, it initiates the now-approved VSC collection.

3 Requesting a Derived Credential

You can request derived credentials for your own mobile device or PC.

Collecting a mobile ID may take several minutes, depending on the complexity of the certificates and the speed of your network connection. If the collection fails due to network problems, you are recommended to use the **Cancel Credential** workflow to cancel the mobile ID, then request another mobile ID for the user.

To request a derived credential, open a web browser and navigate to the Start page on the SSRP web server:

```
https://myserver/Start
```

Follow the on-screen instructions.

If you have multiple authentication methods configured, when the user clicks an external authentication method, they are taken to the identification provider's login page or, if the user is already logged in, they may be automatically authenticated and go directly to the Select Credential Profile screen.

If there is a only one option, the user proceeds directly to that option without having to make a selection.

Note: In previous versions of the SSRP, the initial page was `StartPage` rather than `Start`. If you go to the `StartPage` URL, you automatically use the page for client certificate-based derived credentials where you insert your PIV card. If you want to use an OpenID Connect identity provider, you must go to the `Start` URL instead.

3.1 DN format

For client certificate-based derived credentials, if the supplied DN has the EMAIL component as either the first or last component of the DN, this component is removed. The DN must then have the CN as the first or last component to be valid (it may be part of a multi-attribute RDN). If this is not the case, the error `0001` is returned to the user and the supplied DN logged in the audit.

4 Handling revocation

For client certificate-based derived credentials, if the certificate that Derived Credentials are authenticated with is no longer trusted, it is possible to revoke all credentials that were authenticated by that certificate quickly and immediately.

The Derived Credential Notifications Listener web service contains a method named:

```
int CessationOfTrustOfCertificate(string certificateHash);
```

When this method is invoked with the SHA1 hash of the certificate (the certificate thumbprint), all credentials that the certificate authorized are revoked, and notice for the certificates to be revoked is sent to the Certificate Authority. The response from the invocation will indicate how many credentials were revoked. All revocations are audited.

Note: Transfer of Trust and Updating Details are not applicable to the certificate-based Derived Credential issued by SSRP.

See the [Derived Credentials Notifications Listener API](#) document for more information.

5 Multiple credential handling

5.1 Users with multiple DNs

Users who have multiple credentials, each with different DNs but a common UPN, will be have their requests aggregated into a single account. The resultant derived credential will contain details from the UPN as stored in LDAP. This means that a user may get a different DN in their derived certificate than the one in the certificate they present for authentication.

5.2 Users with multiple devices

It is possible for a user to request and collect multiple sets of derived credentials.

Collecting multiple derived credentials to a Windows PC will result in the creation of multiple virtual smart cards, one for each set of credentials.

It is not possible to collect multiple sets of credentials to a mobile device. In this instance, the collection of the second set of credentials will cancel and replace the existing set of credentials. It is not possible to collect credentials to a mobile device that currently holds the derived credentials for another person.

5.3 Archived private key handling

It is possible to share a certificate between multiple devices. This is useful for encryption and decryption purposes.

To achieve this, ensure that a common certificate policy with an archived private key is available to all derived credential profiles. Set this certificate to **Use Existing**:

Required	Certificate Policy Description
<input checked="" type="checkbox"/> ExchangeUserCAArchive on VIN2012R2DC15*	
Action <input type="text" value="Use existing"/>	Number of historic certificates <input type="text" value="2"/>

The first credential issued with this certificate policy will create a new certificate with a new private key. All subsequent issuances will recover that private key and use that instead of creating a new one.

Note: Only certificates initially issued by the SSRP system can be recovered by the SSRP system.

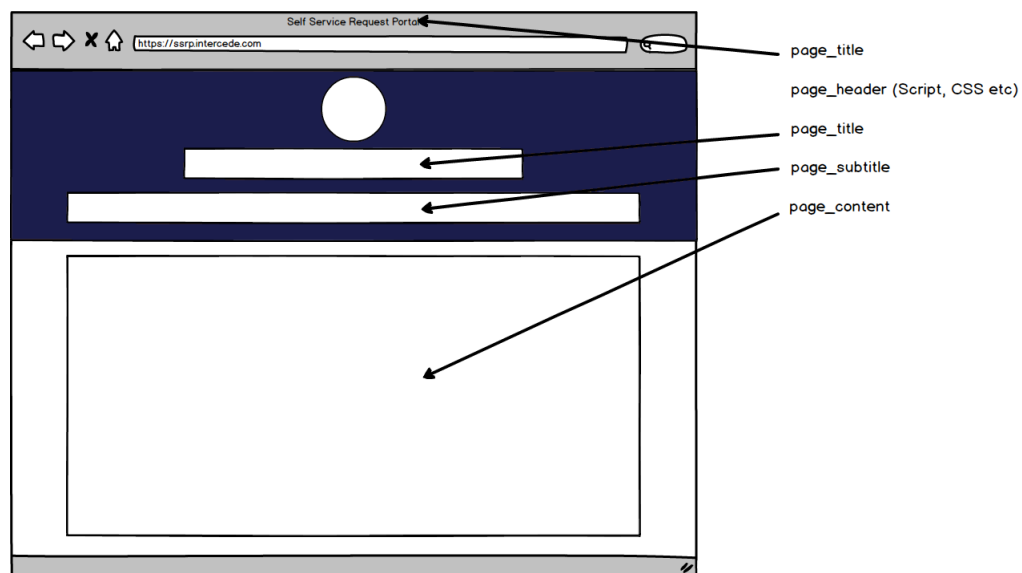
6 Customizing the Self-Service Request Portal

Important: Back up your files before making any changes. You must have a good knowledge of CSS, ASP, and .NET translation before making changes to the SSRP. You are also recommended to back up your files *after* you make your changes, and especially before updating or upgrading your MyID server.

You can customize the appearance of the SSRP by editing the files on the server.

Each page in SSRP uses a template file – `Start.Master`, `StartPage.Master` or `SSRP.Master`. Changes to these files are reflected in all pages on the appropriate site. All pages also use a common CSS file – `css\default.css` – which you can use to change fonts, sizes, and colors.

The page structure is as follows:



Changes you make to the zones in the master pages affect all pages. Changes you make to the zones in the `aspx` pages affect only those pages. For example, if you want to move the logo, or edit the master pages. If you want to add information about the collecting a credential profile, edit the `Instructions.aspx` page.

6.1 Customizing the terminology

Translation is handled using the built-in .Net translation mechanism.

Every text string that appears on the site is stored in the following resource files:

- `App_GlobalResources/StartPage.resx`
- `App_GlobalResources/dictionary.resx`

Changes made to these files are applied at runtime. Any change to this file will automatically cause IIS to restart the website. Modifying these files can also be used for rebranding or providing additional onscreen instructions. The values stored in the `resx` file can contain HTML.

For multilingual support, create a copy of the `resx` file and localize it. Then rename it to have the language and optional culture at the end of the filename, before the extension. For example, the rename the German translated version of `dictionary.resx` to `dictionary.de.resx`, and the US variant to `dictionary.en-us.resx`.

Section	File	Strings
Welcome screen	StartPage/default.aspx	ID0002 ID0003
Help with mutual SSL on Chrome	StartPage/Help_chrome.ascx	Chrome01 - Chrome09
Help with mutual SSL on Edge	StartPage/Help_edge.ascx	Edge1 - Edge09
Help with mutual SSL on Internet Explorer	StartPage/Help_IE.ascx	IE01 - IE10
Help with mutual SSL failure (http:403.4 error)	StartPage/Errors/403_4.aspx	ID0023 - ID0024
Help with mutual SSL failure (http:403.7 error)	StartPage/Errors/403.aspx	ID0011 - ID0015
Help with mutual SSL failure (http:403.13 error)	StartPage/Errors/403_13.aspx	ID0019 - ID0021
Help with mutual SSL failure (http:403.16 error)	StartPage/Errors/403_16.aspx	ID0025 - ID0027
Help collecting Identity Agent	SSRP/Instructions.aspx	ID1009
Help collecting VSC	SSRP/Instructions.aspx	ID1007
Help downloading Android app	SSRP/Download_android.ascx	Android1 Android2 Android3
Help downloading iOS App	SSRP/Download_iOS.ascx	iOS1 iOS2 iOS3
Tab label - the label of the SSRP browser tab.		Start0001
Page Header - the header of the SSRP website under normal conditions.	Start/default.aspx	Start0002
Page description - the description of the SSRP website.	Start/default.aspx	Start0003

Section	File	Strings
Error header - the header of the SSRP website when there is an error.	Start/Error.aspx	Start0004
Error code - this will be followed by an appropriate error code number.	Start/Error.aspx	Start0005
Notification that validation of request is required	SSRP/Instructions.aspx	ID1008

6.1.1 Providing information on the issuance process

If a request for a derived credential does not require validation, SSRP displays a link to initiate the collection of the VSC, smart card, or USB token at the end of the request. If you do not want to display this link, you can suppress the link by clearing the following strings:

- ID1028 – for VSCs
- ID1030 – for smart cards and USB tokens

If you want to provide additional information about the issuance process, for example if an email is being sent to the user with a collection link, you can add this information to the following strings:

- ID1009 – for mobile
- ID1027 – for VSCs that require an auth code
- ID1007 – for VSCs that do not require an auth code
- ID1029 – for smart cards and USB tokens

6.2 Customizing locations

You may want or need to customize the locations that are used by the SSRP. These are located in the `Start/App_GlobalResource/Dictionary.aspx` file within the SSRP parent folder. By default, the SSRP parent folder is located at:

`C:\Program Files\Intercede\MyID\SSRP`

Label	Description
SSRPUrls	A list of URLs of the SSRP instances installed on the server, separated by pipes. These can be full URLs, or just the path from the SSRP parent folder. For example: <code>/SSRP /SSRPOID</code>
StartPageUrl	The URL of the client certificate authentication only SSRP page.
StartUrl	The URL of the main SSRP page.

6.3 Customizing the properties file

There is a separate `myid.json` properties file for each website folder; by default, SSRP for client certificate-based derived credentials, and SSRPOID for OpenID derived credentials.

The default location of the `myid.json` file for the SSRP website folder is:

```
C:\Program Files\Intercede\MyID\SSRP\SSRP\
```

Note: As updating or upgrading MyID overwrites the `myid.json` configuration files, you are recommended to make all of your customizations in a `myid.production.json` file, which is an override file for the `myid.json` file. If you do not have a `myid.production.json` file in the same folder as the `myid.json` file, you can make a copy of the `myid.json` file and rename it to `myid.production.json`. If there is an entry in the `Providers` array with the same `Name` attribute in both the `myid.json` and `myid.production.json` file, the entry in the `myid.production.json` file takes precedence.

For information on configuring the properties file for OpenID, see section [7.3, Configuring the Self-Service Request Portal for external identity providers](#).

6.3.1 Editing the provider properties

For client certificate-based derived credentials, the properties file can contain the following:

```
{
  "Providers": [
    {
      "Name": "SSL",
      "DisplayName": "Login with your PIV Card",
      "Enabled": true,
      "Type": "ssl",
      "Default": true,
      "Icon": "..\\SSRP\\Images\\smartcard.png"
    }
  ]
}
```

You can add or edit the following:

- `DisplayName` – The display name for the provider. This is the text that appears on the start screen.

For example: Login with your PIV Card

- `Enabled` – Whether the provider can be used. If you set this value to `false`, the provider does not appear in the SSRP selection screen.

You can set this property to `true` or `false`. The default value for this property is `false`.

- `Icon` – The file path to the icon of the provider. This is an optional property that, if populated, displays the icon to the left of the display name in the start screen.

For example:

```
..\\SSRP\\Images\\smartcard.png
```

Note: Do not edit the `Name`, `Type`, or `Default` settings for the client certificate-based derived credentials provider.

6.3.2 Customizing the Self-Service App links

You can edit the `myid.json` properties file for each SSRP website folder to specify the MyID CMS server to use for the Self-Service App, and also whether to embed (and optionally display) authentication codes.

The default `myid.json` file contains a commented-out `LaunchClients` section that you can use as a basis for your changes.

Add the following to your `myid.json` file:

```
"LaunchClients": [  
  {  
    "Name": "SSA",  
    "Server": "https://myid.intercede.com",  
    "EmbedAuthCode": true,  
    "ShowEmbeddedAuthCode": true  
  }  
],
```

You can add or edit the following:

- **Name** – Must be `SSA`.
- **Server** – The address of the MyID CMS web server.
This address must also be on the list of allowed servers in the Self-Service App configuration file; see the *Server location* section in the [Self-Service App](#) guide for details.
- **EmbedAuthCode** – Set to `true` to embed the authentication code in the URL used to launch the Self-Service App.
This assumes that the credential profile you are using for the device requires an authentication code.
Note: Embedding the authentication code in the URL for the Self-Service App is supported in the versions supplied with MyID CMS 12.14 onwards. Previous versions of the Self-Service App do not support embedding the authentication code in the URL.
- **ShowEmbeddedAuthCode** – Set to `true` to display the authentication code on screen. You must *also* set `EmbedAuthCode` to `true`.
This option allows you to display the authentication code on screen so that users who have an older (pre-MyID 12.14) version of the Self-Service App installed can collect their device. If all of your end users have the newer version of the Self-Service App installed, you can set this option to `false`.

7 External identity providers

The Self-Service Request Portal supports OpenID Connect identity providers for derived credentials – you authenticate to an external identity provider, and SSRP generates derived credentials based on the claims returned by the external system.

You can configure SSRP for multiple OpenID Connect providers; SSRP provides a choice of providers to the user when they access the SSRP website. You can also configure SSRP for one or more OpenID Connect providers in addition to the SSL provider.

As with standard client certificate authentication, MyID attempts to find a given user in the database and can make changes to the user. In the case of OpenID Connect, this is using the claims returned from the OpenID Connect provider. If the user is in the database, then some fields may be updated with values from the claims. If the user is not in the database, then the user is added using the values from the claims and an attempt is made to look the user up in the Active Directory to pull their details into the database.

If any error is thrown with either the import of the user or the request of the derived credential, then the user is not imported into MyID and no request is created.

Important: Back up your files before you make any changes. You are also recommended to back up your files *after* you have made your changes, and especially before updating or upgrading your MyID server.

To set up external credentials:

- If you want to use both PIV card client certificates and OpenID Connect authentication, set up your SSRP website for multiple authentication provider types.
See section [7.1, Configuring multiple authentication provider types](#).
- Configure the external identity provider so that it can connect to the SSRP.
See section [7.2, Configuring your external identity provider](#)
- Configure the SSRP for the new external identity provider.
See section [7.3, Configuring the Self-Service Request Portal for external identity providers](#).
- For a starting point for Microsoft Entra as an external identity provider, you can use the sample provided.
See section [7.4, Sample configuration for Entra](#).

7.1 Configuring multiple authentication provider types

By default, SSRP is set up for client certificate derived credentials.

You can additionally configure SSRP to use OpenID Connect derived credentials.

The installation process creates the following websites:

- SSRP – contains configuration for the client certificate-based derived credentials.
Located in the following folder by default:
`C:\Program Files\Intercede\MyID\SSRP\SSRP`
- SSRPOID – allows you to add OpenID Connect authentication.

Located in the following folder by default:

```
C:\Program Files\Intercede\MyID\SSRP\SSRPOID
```

If you want to allow people to choose between client certificate derived credentials and OpenID Connect derived credentials, you configure the `SSRPOID` website folder for the SSRP web service for OpenID Connect authentication; you can configure each SSRP website folder for *either* client certificates *or* OpenID Connect, but not both. Note, however, that you can include multiple OpenID Connect identity providers in the same SSRP website folder.

Note: In previous versions of the SSRP, the initial page was `StartPage` rather than `Start`. If you go to the `StartPage` URL, you automatically use the page for client certificate-based derived credentials where you insert your PIV card. If you want to use an OpenID provider, or to offer a choice between client certificate-based derived credentials and OpenID Connect derived credentials, you must go to the `Start` URL instead.

Important: If you have a requirement to issue PIV Derived Credentials in accordance with US Government standard NIST SP-800-157, you must ensure that the PIV Derived Credential profile is not available to users who authenticate with an OpenID Connect Identity provider. It is recommended that distinct roles are assigned for each credential profile to ensure client certificate derived credentials are used.

7.2 Configuring your external identity provider

How the external identity provider works may vary between providers. You must set up the following in the external identity provider:

- The client ID.

This is an identifier for the OpenID Connect application. Create this in your identity provider.

- A client secret.

This is a password that secures the connection between SSRP and the identity provider. Create a new secret in your identity provider. Make sure you take a note of the secret when you create it; for security reasons, typically you cannot recover a client secret after you have created it and navigated away from the creation screen.

- Redirect URI.

You must configure the identity provider to allow responses to be returned to SSRP.

Add the following to the external provider's list of allowed redirect URIs:

```
https://<server><website>?name=<name>
```

For example, you have a website called:

```
myserver.example.com
```

Your configured web application for OpenID Connect identity providers has the alias:

```
/SSRPOID
```

In the `myid.production.json` file for this website folder, you have set the `Name` attribute to:

```
Entra
```

When you set up your external identity provider, you must allow the following redirect URI:

`https://myserver.example.com/SSRPOID?name=Entra`

7.3 Configuring the Self-Service Request Portal for external identity providers

You configure the identity providers for SSRP by editing the `myid.production.json` file in the website folder, which is an override for the `myid.json` configuration file. There is a separate `myid.json` file for each website folder; by default, SSRP for client certificate-based derived credentials, and SSRPOID for OpenID Connect derived credentials.

Note: As updating or upgrading MyID overwrites the `myid.json` configuration files, you are recommended to make all of your customizations in a `myid.production.json` file, which is an override file for the `myid.json` file. If you do not have a `myid.production.json` file in the same folder as the `myid.json` file, you can make a copy of the `myid.json` file and rename it to `myid.production.json`. If there is an entry in the `Providers` array with the same `Name` attribute in both the `myid.json` and `myid.production.json` file, the entry in the `myid.production.json` file takes precedence.

The `myid.production.json` file defines each potential identity provider within that overall identity provider type from which SSRP can create the new credential. The file contains the URLs and data for authentication with external authentication providers, and the mappings from the claims to the database.

You can include multiple providers in a single `myid.production.json` file if the providers are all the same type; for example, you can include multiple OpenID Connect providers in the same file. You cannot include a client certificate-based provider in the same `myid.production.json` file as OpenID Connect providers.

The default location of the `myid.json` file (and `myid.production.json`) for OpenID Connect is:

```
C:\Program Files\Intercede\MyID\SSRP\SSRPOID
```

By default, the `myid.json` file contains a copy of the `myid.json` file from the SSRP folder. You must copy this into a `myid.production.json` file, then edit this file to remove the client certificate-based provider (with a `Name` of `SSL` and a `Type` of `ssl`) and configure the file for OpenID Connect authentication.

The format of the `myid.production.json` is:

```
{
  "Providers": [
    {
      "Name": "<The internally used name for the identity provider>",
      "DisplayName": "Text displayed to the user",
      "Url": "<URL of OpenID provider>",
      "Icon": "<path to icon>",
      "Enabled": true,
      "Type": "oid",
      "Default": true,
      "Scopes": "openid email profile",
      "RequiredAudience": "<Required Audience of the JWT>",
      "RequiredIssuer": "<Required Issuer of the JWT>",
      "ClientId": "<Client ID>",
      "ClientSecret": "<Client Secret>",
      "RetrieveUserInfo": true,
      "Mappings": [
```

```

{
  "Match": {
    "<Type of Claim>": "<Required Value of Claim>"
  },
  "Attributes": [
    {
      "From": "preferred_username",
      "To": "Email",
      "Mandatory": false,
      "Default": "",
      "Static": "",
      "Unique": true,
      "Update": false,
      "LdapSync": true
    },
    {
      "From": "",
      "To": "Roles",
      "Static": "<role name='Derived Credential Owner' scope='1'/><role
name='Cardholder' scope='1'/>"
    },
    {
      "From": "",
      "To": "GroupName",
      "Static": "Imported From Google"
    }
  ]
}
]
}
]
}

```

Note: When editing the `appsettings.Production.json` file on your own system, make sure you replace any placeholders with the appropriate values for your system.

Each provider within the `Providers` array can contain:

- **Name** – An internal name for the provider. When configuring the OpenID Connect provider, the redirect URIs must contain:

{URL of SSRP instance}?name={Name in myid.production.json}

For example, if the **Name** is `Entra`, the redirect URI might be:

`https://myserver.example.com/SSRPOID?name=Entra`

- **DisplayName** – The display name for the provider. This is the text that appears on the start screen.

For example: `Login with your Microsoft account`

- **Url** – The URL of the OpenID Connect provider.

For example:

`https://login.microsoftonline.com/b785ece2-47dd-4eb6-acee-be595bbce9b3/v2.0`

Note: With some providers, such as Microsoft Entra, you can type this URL into a browser and append:

`/.well-known/openid-configuration`

to return JSON that contains all the relevant information required to authenticate with the provider.

- **Icon** – The file path to the icon of the provider. This is an optional property that, if populated, displays the icon to the left of the display name in the start screen.

For example:

`..\SSRPOID\Images\Microsoft.png`

- **Enabled** – Whether the provider can be used. If you set this value to `false`, the provider does not appear in the SSRP selection screen.

You can set this property to `true` or `false`. The default value for this property is `false`.

- **Type** – The type of authentication that is used to request the derived credential. Set this to `oid` for OpenID Connect authentication.
- **Default** – Whether this provider is the default if a user navigates directly to the SSRP URL. You can set this to `true` for at most one provider.

If multiple providers have this set to `true`, then only the first provider in the list from top down that is set to `true` is used. If there are no providers set to `true`, then the first provider is used. When going to the `Start` page, this setting is ignored.

You can set this property to `true` or `false`. The default value for this property is `NULL`. When this property is set to `NULL`, it is ignored.

- **Scopes** – The scopes requested when attempting to authenticate with the OpenID Connect provider. Each scope allows for a different set of data to be returned in the claims, and typically `openid` is always required. See the JSON returned from the OpenID Connect provider's `/.well-known/openid-configuration` URL, or the OpenID Connect provider's documentation for a list of the allowed scopes and claims.

You must set this property to a string with each scope separated by a space; for example:

`"Scopes": "openid email profile"`

- **RequiredAudience** – The required audience. This is checked against the value of the audience of the JSON Web Token (JWT) returned from the OpenID Connect provider after the user has authenticated. If this is empty or not present, the audience of the JWT is not checked. If the value of this property does not match the audience of the JWT, an error occurs.
- **RequiredIssuer** – The required issuer. This is checked against the value of the required value of the issuer of the JSON Web Token (JWT) returned from the OpenID Connect provider after the user has authenticated. If this is empty or not present, the issuer of the JWT is not checked. If the value of this property does not match the issuer of the JWT, an error occurs.
- **ClientId** – The client ID as configured in the OpenID Connect provider. When you configure the OpenID Connect provider, you are given a client ID and client secret. This property must contain the client ID.
- **ClientSecret** – An encrypted version of the client secret as configured in the OpenID Connect provider. When configuring the OpenID Connect provider, you are given a client ID and a client secret.

Note: You are recommended to encrypt the client secret for security purposes; see section 7.3.1, *Encrypting the client secret*.

- `ClientSecretClear` – The non-encrypted version of the client secret.

Note: For evaluation systems, if you do not want to go to the effort of encrypting the client secret, you can leave the `ClientSecret` value blank, and set the `ClientSecretClear` value to the client secret instead. This is not recommended on production systems for security reasons.

- `AuthorizationUrl` – The URL of the authorization endpoint for the OpenID Connect provider.

For example:

```
https://login.microsoftonline.com/b785ece2-47dd-4eb6-acee-be595bbce9b3/oauth2/v2.0/authorize
```

Note: If the OpenID Connect provider has a `/.well-known/openid-configuration` endpoint that returns JSON, then this may not be required, as the URL can be retrieved from the JSON. The property name in the JSON returned from the OpenID Connect configuration endpoint is:

```
authorization_endpoint
```

- `TokenUrl` – The URL of the token endpoint for the OpenID Connect provider.

For example:

```
https://login.microsoftonline.com/b785ece2-47dd-4eb6-acee-be595bbce9b3/oauth2/v2.0/token
```

Note: If the OpenID Connect provider has a `/.well-known/openid-configuration` endpoint that returns JSON, then this may not be required, as the URL can be retrieved from the JSON. The property name in the JSON returned from the OpenID Connect configuration endpoint is :

```
token_endpoint
```

- `SigningKeysUrl` – The URL of the signing keys endpoint for the OpenID Connect provider.

For example:

```
https://login.microsoftonline.com/b785ece2-47dd-4eb6-acee-be595bbce9b3/discovery/v2.0/keys
```

Note: If the OpenID Connect provider has a `/.well-known/openid-configuration` endpoint that returns JSON, then this may not be required, as the URL can be retrieved from the JSON. The property name in the JSON returned from the OpenID Connect configuration endpoint is:

```
jwks_uri
```

- `RetrieveUserInfo` – Whether to retrieve more information on an authenticated user. If you set this to `true`, then once the user has authenticated, an additional endpoint call is made to the OpenID Connect provider user info endpoint to retrieve more information about the user. Use this option if the returned claims do not provide sufficient information.

You can set this property to `true` or `false`. The default value for this property is `false`.

- `UserInfoUrl` – The URL of the user info endpoint at the OpenID Connect provider.

For example:

```
https://graph.microsoft.com/oidc/userinfo
```

Note: If the OpenID Connect provider has a `/.well-known/openid-configuration` endpoint that returns JSON, then this may not be required, as the URL can be retrieved from the JSON. The property name in the JSON returned from the OpenID Connect configuration endpoint is:

```
userinfo_endpoint
```

This is used only if `RetrieveUserInfo` is set to `true`.

- **Mappings** – An array of mappings that contains the following:
 - **Match** – A set of key-value pairs on which to match. For each key-value pair, the key must be a type of claim (for example: `"iss"` or `"aud"`) and the value is the required value for that claim. If there are multiple key-value pairs, then every key-value pair must match a claim, and if there are no key-value pairs, or if the match property is not present, then the match is always successful. If multiple **Mappings** have every item in their **Match** property met, then the first successful mapping in the mappings array is used. If no **Mappings** have successfully matched **Match** properties, then an error occurs.
 - **Attributes** – An array of items that define how the user is created, imported, or updated in the MyID database. Each attribute defines what information to get from where. Each attribute can contain the following properties:
 - **From** – The type of claim from which to obtain the value.
For example: `"iss"` or `"aud"`.
If you leave this empty, then the **Static** property is used. If this returns nothing, this either causes an error, or the value in the **Default** property is used, depending on the value of the **Mandatory** property.
 - **Static** – A static value to send to MyID. This is used only if the **From** property is empty or not present.
 - **Mandatory** – Whether a response from the claim specified in the **From** property is necessary. If you set this to `true` and there is either no claim for the type specified in the **From** property, or if the value of the claim is empty, then an error occurs.
You can set this property to `true` or `false`. The default value for this property is `false`.
 - **Default** – The default value to use. If the **Mandatory** property is set to `false` and either there is no claim for the type specified in the **From** property, or if the value of the claim is empty, then the value from this property is used instead.
 - **To** – The name of the node in the XML sent to MyID. This must match a column name in the `vPeopleUserAccounts` view in the MyID database.

To add user to a group, you must add an attribute with the **To** property set to `"GroupName"`, and the **From/Static/Default** property set to the name of the group.

If the group you are adding the user to is a sub-group, add another attribute with the `To` property set to `"ParentGroupName"` and the `From/Static/Default` property set to the name of the parent group.

To assign roles to a user, you must add an attribute with the `To` property set to `"Roles"` and the `From/Static/Default` property set to XML of the role or roles which you want to add.

For example, you could set the `To` property to `"Roles"` and the `Static` property to:

```
"<role name='Derived Credential Owner' scope='1' /><role  
name='Cardholder' scope='1' />"
```

This gives the user two roles, "Derived Credential Owner" and "Cardholder", both with scope "Self".

The numbers correlate to the scopes as follows:

- 1 – Self
- 2 – Department
- 3 – Division
- 4 – All

For more information on scopes, see the *Scope and security* section in the [Administration Guide](#).

Note: If the user does not already exist in the database, there must be an attribute with `To` set to `"FullName"` and the `From/Static/Default` property set to the user's full name. If this is not present, the creation of the user fails.

- **Unique** – Set this to `true` if the `To` and `From/Static/Default` properties lead to a unique value that can be used to identify the person. If you set this to `true`, then the `vPeopleUserAccounts` view in the MyID database is searched using the field specified in the `To` property and the value specified in the `From/Static/Default` property. If a user is found using these criteria, then that user and only that user is imported. There must be at most one attribute where this property is set to `true`. This must reference the Unique ID of the user in the external identity provider.

You can set this property to `true` or `false`. The default value for this property is `false`.

You are recommended to use a unique reference (such as a GUID) supplied from the external identity provider; you can store this value in a MyID attribute to create a link between the user in the external identity provider and MyID.

MyID provides the following fields:

- `XuSYSExternalReferenceId1`
- `XuSYSExternalReferenceId2`
- `XuSYSExternalReferenceId3`

that you can use to store the unique ID. If you have up to three external identity providers, you can use a different field for each.

For example:

```
{
  "From": "oid",
  "To": "XuSYSExternalReferenceId1",
  "Mandatory": true,
  "Unique": true,
  "Update": true
},
```

- **Update** – Whether the user in the database should be updated to contain information from this attribute. If you set this to `true`, the user in the database (who was found using either an attribute that you have stated as unique or one or more attributes with `LookupExisting` set to `true`) is updated to have the field specified in the `To` property set to the value specified in the `From/Static/Default` property.

You can set this property to `true` or `false`. The default value for this property is `false`.

- **LookupExisting** – If you set this to `true`, then, after the attribute with `Unique` set to `true` is used to search the database for the user, if no user is returned, then instead `vPeopleUserAccounts` is searched using the field specified in the `To` property and the value specified in the `From/Static/Default` property of this attribute. If the search using the value of the unique attribute returns a single result, then this is not used. A unique attribute should not return more than one result after being searched for in a database.

If multiple attributes have `LookupExisting` set to `true`, the user must have all of those properties set to the required values.

You can set this property to `true` or `false`. The default value for this property is `false`.

- **LdapSync** – Whether to search the LDAP for this user. If you set this to `true`, then the linked Active Directories are searched using the field specified in the `To` property and the value specified in the `From/Static/Default` property. The value of the `To` property must be a value in the `LDAPLookUp` table.

You can set this property to `true` or `false`. The default value for this property is `false`.

Important: If you enable `LookupExisting` or `LdapSync` on an attribute, you must be certain that the source of that data from the external identity provider is trustworthy. If you use these features and the source of the mapped attribute used for `LookupExisting` or `LdapSync` can be controlled by the end user or another untrusted individual, it can enable the user authenticating with that identity provider to impersonate a user in MyID, either by assigning the external identity provider authentication mechanism to that existing user account or by importing data (such as the DN) from the LDAP that belongs to another person who is looked up by that attribute.

For a sample `myid.production.json` file containing configuration for Microsoft Entra, see section 7.4, [Sample configuration for Entra](#).

7.3.1 Encrypting the client secret

To encrypt the client secret, log in to the server with the MyID Web Service account and DPAPI encrypt the secret.

1. On the web server where your SSRP web service is located, log in to the server with the MyID Web Service account.

To confirm which account to use, check the settings for the `SSRPPOOL` application pool in IIS. You must log in with the same account used to run the web service, or the web service will be unable to decrypt the client secret.

2. Open a Windows PowerShell command prompt, and navigate to the folder where `myid.production.json` is located.
3. Run the following PowerShell script:

```
.\DPAPIEncrypt.ps1 <secret>
```

where:

- `<secret>` is the client secret from when you configured your OpenID Connect provider.

For example:

```
.\DPAPIEncrypt.ps1 b5989015-bb9e-4533-874b-2b4a6a8280ed
```

The script outputs an encrypted copy of the secret; for example:

```
PS C:\Program Files\Intercede\MyID\SSRP\SSRP> .\DPAPIEncrypt.ps1  
b5989015-bb9e-4533-874b-2b4a6a8280ed  
AQAAANCMnd8BFdERjHoAwE/C [...] JwWwaKXWoS3i+ulxtmjVQyudpQ==
```

(Encrypted output string truncated for documentation purposes.)

4. Copy the encrypted secret, and paste it into the `ClientSecret` property of the `myid.production.json` file.

7.4 Sample configuration for Entra

This section contains a sample `myid.production.json` file set up for Microsoft Entra that you can use as a starting point for your own configuration. Copy this information into the `myid.production.json` file in the SSRPOID folder. You may have to create this file, or replace the contents of an existing `myid.production.json` file; by default, the SSRPOID folder is in the following location:

```
C:\Program Files\Intercede\MyID\SSRP\SSRPOID
```

7.4.1 Configuring the file for your own settings

You must replace the following settings with your own values:

- `Url` – replace the tenant ID in the middle of the URL with your own tenant ID.
`https://login.microsoftonline.com/<tenant ID>/v2.0`
You can obtain your tenant ID from the Microsoft Entra Overview page.
- `ClientId` – replace with your own client ID.
You can obtain your client ID from the Microsoft Entra Overview page.
- `ClientSecret` – replace with an encrypted version of your own client secret.
You can obtain your client secret when you create it in the Microsoft Entra portal, and you can encrypt it using the provided `DPAPIEncrypt.ps1` PowerShell script; see section [7.3.1, *Encrypting the client secret*](#) for details.

See section [7.3, *Configuring the Self-Service Request Portal for external identity providers*](#) for more information about the content of this file.

7.4.2 Setting up redirect URIs

You must also make sure that you add the Redirect URI for the provider in the Microsoft Entra portal.

In this case, the `Name` is `Entra`, and it is included in the `SSRPOID` web application, so if your server name is `myserver.example.com`, the redirect URI would be:

```
https://myserver.example.com/SSRPOID?name=Entra
```

See section [7.2, *Configuring your external identity provider*](#) for details.

7.4.3 Example myid.production.json file for Microsoft Entra

```
{
  "Providers":[
    {
      "Name":"Entra",
      "DisplayName":"Login with your Microsoft account",
      "Url":"https://login.microsoftonline.com/b785ece2-47dd-4eb6-acee-be595bbce9b3/v2.0",
      "Enabled":true,
      "Type":"oid",
      "Default":true,
      "Scopes":"openid email profile",
      "RequiredAudience":"",
      "RequiredIssuer":"",
      "ClientId":"4d58a40d-d199-46d7-9da5-f363b071fc44",
      "ClientSecret":"AQAAANCmnd8BFdERjHoAwE/C [...] JwWwaKXWoS3i+ulxtmjVQyudpQ==",
      "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":"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'/>"
            },
            {
              "From":"",
              "To":"GroupName",

```

```
        "Static": "Imported From Microsoft"
      },
      {
        "From": "",
        "To": "ParentGroupName",
        "Static": "Derived Credentials"
      }
    ]
  }
}
```

8 Error codes and logging

This section provides information about the error codes that appear when using the SSRP, and details of how to enable logging.

To enable logging, edit the `log.config` file in the SSRP or SSRPOID website. Change the log level from `OFF` to `ALL`.

You can also change the logging path; you are recommended to use a path that is not protected by the Windows UAC settings. For example:

```
C:\Logs\SSRP.log
```

Important: Enabling these logs captures data that is sent to the application server. These logs may contain sensitive data, and generating the logs may impact performance. Enable these logs only when diagnosing issues, and ensure that the log files created are sufficiently protected on the server.

See section [1.4, What information is stored in MyID when I request a Derived Credential?](#) for details about the type of information that may be present.

8.1 Error code reference

Number	Name	Description
0001 E0001 ED0001	Unknown	<p>Something unexpected has occurred.</p> <p>Review the MyID Audit Reporting and System Events workflows for further details.</p> <p>Possible causes include the MyID License limit being reached or attempting to add a new user when this feature is disabled.</p> <p>This may also occur when no SSRP URLs have been configured. The value for the key <code>SSRPUrls</code> does not contain any valid SSRP URL.</p> <p>Enabling logging and repeating the process may help diagnose the issue.</p>
E0002 ED0002	Unknown	<p>No Identity Providers are enabled. The <code>myid.json</code> or <code>myid.production.json</code> files for each SSRP instance have been set up with every identity provider disabled.</p>
0002	NoClientSSLCertificate	<p>The SSRP site is not configured to accept client certificates, the client lacks an appropriate certificate, or the client is not providing a certificate.</p> <p>Check the SSRP SSL settings.</p>

Number	Name	Description
0003	NoSuitableCredentialProfiles	<p>The certificate that the user is presenting does not qualify for any credential profiles.</p> <p>If the user is trying to use client certificate authentication and you are expecting them to qualify for a credential profile, check the roles being assigned in the configuration file (as defined in section 2.8.8, Configuring the available credential profiles), then ensure that the credential profile has a capability of Derived Credential and that these roles allowed for Can Request, Can Collect, and Can Receive.</p> <p>If the user is trying to use OpenID authentication, check that the roles that you are adding using the <code>myid.production.json</code> file allow Can Request, Can Collect, and Can Receive.</p>
0004	SpecifiedProfileNotAllowed	<p>An attempt has been made to request a credential profile that the certificate holder is not allowed to receive.</p> <p>Having the certificate holder browse to the StartPage site will present a list of all the profiles that they can receive.</p>
0005	TooManyCredentialProfiles	<p>There are multiple profiles available to the user. This error is always caught and handled, so should never be presented to the user.</p>
0006	UnrecognisableJobStatus	<p>The Derived Credential request has been created with an unexpected status.</p> <p>Enabling logging and repeating the process may help diagnose the issue.</p>

Number	Name	Description
0007	UserNotFound	<p>The Derived Credential request has not matched an existing account, and account creation is disabled in the configuration.</p> <p>See section 2.4, MyID configuration options for further details.</p>
0008	LicenseExpired	<p>Your MyID license has expired. Contact Intercede to arrange a new license.</p> <p>The text presented to the user for this message is deliberately generic.</p>
0009	CardProfileRequisiteDataCheckFailed	<p>The request cannot be created because required information is not available.</p> <p>The credential profile used has Requisite User Data set, but the person does not have the required data; check the audit to see what credential profile was selected, and what required data was missing.</p>
0010	UserCannontBeImportedDueToRoleRestrictions	<p>The role specified in the <code>ssrp.conf.xml</code> file can not be given to a user in the group to which the user is being attempted to be added.</p>

Number	Name	Description
0011	ConfigurationError	There is an error in the <code>ssrp.conf.xml</code> file; check that the <code>userprofileid</code> is correct and present in the database.
0012	AuthenticationFailed	The attempt to authenticate using the OpenID provider has failed. To help diagnose the issue, enable logging and repeat the process. There will not be anything in <code>LogEvents</code> OR <code>SystemEvents</code> .
0013	InvalidPermissionsForCredential	The user has successfully authenticated using the OpenID provider, but the user they are authenticated as lacks the required permissions for the credential. Check your configuration of the <code>myid.production.json</code> file. The issue is likely to be in the <code>Mappings</code> or, within that, the <code>Attributes</code> section. To help diagnose the issue, enable logging and repeat the process. There will not be anything in <code>LogEvents</code> or <code>SystemEvents</code> .

In addition to this, Mutual SSL failure errors (HTTP 403.4, 403.7, 403.13 and HTTP 403.16) and malformed URL errors (HTTP 404) are handled by custom error pages. Additional details can be added to these as appropriate.

8.1.1 SSL authentication error

When the browser requests an SSL certificate, if you select a valid certificate but then click **Cancel** on the PIN dialog, the browser displays a generic SSL authentication error instead of a custom SSRP error page. Close all instances of the browser and try to carry out the operation again.

8.2 Troubleshooting mobile device collection

If you are using the **Show app collection link on mobile OS** configuration option in the credential profile to allow users to request and collect a derived credential directly on a mobile device, and you experience problems where the app collection link does not appear, check the **Audit Reporting** workflow for the event when the request is created. Look for the browser's `UserAgent` and confirm whether it contains either `iOS` or `Android`; if it does not, SSRP cannot determine that the device is a mobile device, and so does not display the app collection link.

9 Mutual SSL behavior

For client certificate-based derived credentials, SSRP relies on using Mutual SSL for user identification. Browsers handle Mutual SSL differently, and often counter intuitively.

This behavior should be considered, including impact on usability, before deploying SSRP, and consideration given to updating help pages with instructions for dealing with common problems in your environment; see section [6.1, Customizing the terminology](#).

This section is not relevant for OpenID-based derived credentials.

9.1 Caching credentials

An important behavior common to Chrome, Internet Explorer, and Edge is that once a client certificate has been selected, the certificate will remain the choice for that URL until all instances of that browser have been closed. Closing the authenticated tab, or even removing the smart card containing the certificate, does not end the trusted session.

This equally applies to choosing no certificate (clicking Cancel on the certificate dialog). The only way to recover from this is to close all instances of that browser, then try again. Browser based extensions, such as media controls, also need to be closed. This is a very important consideration if the connecting client is on a shared PC.

9.2 Persisting credentials

By default, Windows will import the certificates from any smart card it sees. This means that when a user is offered a certificate to select, they will be presented with a list of every certificate that has ever used on that Windows account on that PC. If this is a shared PC, that may be seen as a privacy concern. At a minimum, it means the user may have to pick their certificate from a long, unordered list. If a user is using their own Windows credentials, this is not a problem.

9.3 Certificate selection hidden from user

Internet Explorer has been observed to have its certificate prompt appear behind the browser. This makes it appear that the browser has stopped responding. A note has been added to the start page to make users aware of this behavior.

9.4 Smart card not registered

It is possible for Windows to stop propagating certificates for a specific smart card. This means that the certificates for the smart card will not be available to the browser. To resolve this issue, restart Windows.