MyID PIV

Version 12.14

Web Service Architecture

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

Document reference: INT1973-12.14.0-PIV

March 2025

Copyright

Conventions used in this document

- Lists:
  - Numbered lists are used to show the steps involved in completing a task when the order is important.
  - Bulleted lists are used when the order is unimportant or to show alternatives.
- **Bold** is used for menu items and for labels.

  For example:
  - Record a valid email address in **'From' email address**.
  - Select **Save** from the **File** menu.
- *Italic* is used for emphasis:

  For example:
  - Copy the file *before* starting the installation.
  - Do *not* remove the files before you have backed them up.
- ***Bold and italic*** hyperlinks are used to identify the titles of other documents.

  For example: "See the ***Release Notes*** for further information."

  Unless otherwise explicitly stated, all referenced documentation is available on the product installation media.
- A `fixed width` font is used where the identification of spaces is important, including filenames, example SQL queries and any entries made directly into configuration files or the database.
- **Notes** are used to provide further information, including any prerequisites or configuration additional to the standard specifications.

  For example:

  Note: This issue only occurs if updating from a previous version.
- Warnings are used to indicate where failure to follow a particular instruction may result in either loss of data or the need to manually configure elements of the system.

  For example:

  Warning: You must take a backup of your database before making any changes to it.

Contents

1 Introduction

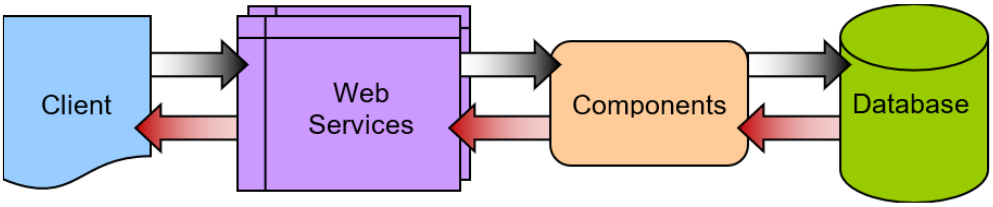
This document describes the MyID® Web Service Architecture. The web services installed on your web server allow MyID end-user applications to communicate with your MyID system. For example, the web services allow you to use the following client applications:

- MyID Desktop
- MyID Self-Service App
- MyID Self-Service Kiosk
- MyID Mobile Identity Management

The following web services are provided:

- MyID Process Driver Web Service – allows a client application to communicate with the MyID application server to carry out card and identity management.
- MyID Data Source Web Service – provides form definitions and device configuration data to a client application. This is a read-only web service with no security restrictions.
- Certificate Check Web Service – allows a client application to check the status of its certificates using the Windows API. This is an optional web service.

1.1 Overview



The client passes requests through HTTP or HTTPS to the MyID Data Source and MyID Process Driver web services; both services are required for full operation. The web services communicate using DCOM with the MyID components on the application server; these components provide the business logic and communicate with the MyID database. Responses are returned to the client through the MyID web services.

The web services, components and database may be on separate servers, or on the same server. The two web services must be installed on the same server.

1.2 Prerequisites

The MyID Web Service Architecture is provided either as a stand-alone update or as part of the main MyID product installation program.

For MyID versions supported and any prerequisites, see the ***Installation and Configuration Guide*** provided on the MyID product media or the `readme.html` document provided with the software update.

In addition:

- You must have .NET 4.8 installed on the server on which the web services are installed.
- Your client applications must be able to communicate over HTTPS to the web server on which you have installed the MyID web services.

  You *must* set up SSL/TLS on this connection.
- The MyID web services must be able to communicate with the MyID components using DCOM. If the web services reside on a separate server to the MyID application server, you must set up the appropriate COM+ proxies.

  See section *2.2.3*, *Setting up the COM+ proxies* for details.

2 Installing the web services

The web services are provided as part of the main MyID product installation program.

To install the web services as part of the MyID installation, in the MyID Installation Assistant, on the Server Roles and Features screen, select the **MyID Client Support\MyID Client Web Service** option.

Note: If you are installing the web services on a separate server to the main MyID web server, you must configure the web services with the location of the MyID web server. See section *2.2*, *Setting up the MyID web services on a standalone server*.

See the *Selecting the server roles and features* section in the ***Installation and Configuration Guide*** for details of the installation procedure.

2.1 Web service configuration

By default, the MyID web services are installed to the following folder:

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

In the root of this folder are folders for each of the individual web services:

- `MyIDDataSource` – contains a `myid.config` file.
- `MyIDProcessDriver` – contains a `myid.config` file.

Note: After making any changes to the `myid.config` files, you must recycle the web service app pool:

1. On the MyID web server, in Internet Information Services (IIS) Manager, select **Application Pools**.
2. Right-click the **MyIDWebService** application pool, then from the pop-up menu click **Recycle**.

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

2.1.1 Session ID setting

The way MyID handles session ID generation was changed in an update for MyID PIV 9.0 SP1. Accordingly, for version 8.0 SP2 systems, make sure the `myid.config` file contains the following line:

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

For all later systems, the value must be:

```
<add key="SessionIDServerGenerated" value="true"/>
```

2.1.2 Configuring self-unlock

To allow an individual server to support self-unlock operations for PIV systems, you must edit the `myid.config` file in the `MyIDProcessDriver` folder. Change the value of the following line:

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

to:

```
<add key="AllowSelfUnlockForPIV" value="true"/>
```

A value of `false` overrides all other settings – if the value is `false` you will not be able to carry out self-unlock operations.

You can use this option to implement different behavior on different servers – for example, the server for attended kiosks could allow self-unlock operations, while the server for an unattended lobby kiosk could prevent these operations.

Note: This feature is available only on servers running the MyID web services version 1.9.1000.1 and later. You do not need to make this change in the `myid.config` of any previous versions.

2.1.3 DN validation

If you see an error similar to the following:

```
410076 - The specified DN is not valid.
```

and you believe the DN *is* valid, you can bypass the DN validation in MyID; edit the `myid.config` file in the `MyIDProcessDriver` folder, and add the following line to the `<MyIDSettings>` section:

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

2.1.4 Rate limiting session count

You can specify a maximum number of sessions for clients to connect to the server. If the number of concurrent sessions exceeds this value, subsequent logon attempts are denied with an error similar to the following:

- `85183 - Server busy, please try again later.`

To set the maximum number of sessions, edit the `myid.config` file in the `MyIDProcessDriver` folder, and add the following line to the `<MyIDSettings>` section:

```
<add key="MaxSessionCount" value="value" />
```

Set the value to the number of sessions; for example, to set a maximum of 100 sessions:

```
<add key="MaxSessionCount" value="100" />
```

To remove the restriction, delete the key from the `myid.config` file, or set the value to `-1` as follows:

```
<add key="MaxSessionCount" value="-1" />
```

Note: If you have multiple servers, you must set this limit on each server. You can use different limits for each server; for example, if you have public-facing servers you may want to limit the number of sessions, while private internal servers are not limited.

2.2 Setting up the MyID web services on a standalone server

You may want to set up your MyID web services on a different server to the MyID application or web servers; in this case, you must carry out some additional configuration.

2.2.1 Configuring the server

For a standalone web services server, follow the instructions in the *Preparing your system* section in the ***Installation and Configuration Guide*** for preparing a system for a web server.

Note, however, that a standalone web services server does not need all of the role services that a web server needs. You must have the following role services:

- Static Content
- Default Document
- ASP.NET
- .NET Extensibility
- ISAPI Extensions
- ISAPI Filters
- Request Filtering
- IIS Management Console

You are also recommended to have the following:

- HTTP Logging

2.2.2 Installing .NET framework

You must install .NET Framework 4.8 on the server.

2.2.3 Setting up the COM+ proxies

If the web services are on a different server to the MyID application server components, you must export the MyID COM+ proxies to the server on which the web services run. This allows the web services to communicate with the MyID COM+ components on the 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
```

You need to install the following proxies:

- APDUCardServer
- Edefice_BOL
- Edefice_CS
- ExpiringItems
- MyIDSCEPHandler (required only if you are using SCEP or iOS OTA)

Different web services require different proxies; see the table below for details.

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 section *3.9, Reverse proxies and load balancing*.

2.2.3.1 Proxies required for each web service

The following table describes which proxies are required for each individual web service:

	APDUCard Server	Edefice_ BOL	Edefice_ CS	Expiring Items	MyIDSCEP Handler
Lifecycle API		✓		✓	
MyID Client Web Service	✓	✓	✓	✓	
Credential Web Service		✓	✓	✓	
Device Management API		✓		✓	
Mobile iOS OTA		✓		✓	✓
Reporting Web Service		✓		✓	
PIV Derived Credentials Notifications Listener	✓	✓		✓	
SCEP API				✓	✓

2.2.4 Installing the MyID web service components

You must install the web services on the server using the supplied installation program. This installer creates the virtual directories and the application pool for the web services.

2.2.5 Setting the location of the web server

If the web services server is not the same server as the web server, you must edit the `myid.config` file in the `MyIDProcessDriver` folder. Add the following line:

```
<add key="WebServer" value="https://myserver"/>
```

Where `myserver` is the domain name of your MyID server. You do not need to include the MyID virtual directory.

Note: The case of `WebServer` is important.

You must also set the **Image Upload Server** configuration option if the web services server is not the same server as the web server.

- On the **Video** page of the **Operation Settings** workflow, set **Image Upload Server** to the name or IP address of the MyID web server. Do not include `http` or `https`, any virtual directories, or any slashes – the IP address or server name are sufficient.

If you do not set this option, some images within MyID will not appear correctly.

- **IKB-50 – Resolving host names**

  When obtaining the images for a card layout, MyID needs to know the location of the server on which the images are stored. The **Image Upload Server** configuration option

contains the name of the server; however, this configuration option may contain an external URL used by clients and may not be resolvable on the MyID server, resulting in missing images.

As a workaround, you can add an entry to the `hosts` file on the server hosting the MyID Web Service.

For example, if the **Image Upload Server** configuration option contains `myserver.example.com`, which *should* resolve to the same server as the MyID Web Service, add the following lines to the following file:

```
C:\Windows\System32\drivers\etc\hosts
```

```
127.0.0.1   myserver.example.com
::1         myserver.example.com
```

2.2.6 Troubleshooting

If you have an existing server which has .NET 4.8 and IIS already installed and the site is not working as expected, try running the following statement at the Windows command line:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\ aspnet_regiis.exe -i
```

This command ensures that .NET 4 is registered with IIS.

2.3 Configuring the MyID web services for Integrated Windows Logon

If you set up the MyID server to use Integrated Windows Logon, some applications using the web services can use the cardholder's currently logged-on Windows identity to authenticate to MyID without having to enter passphrases or use a smart card.

See the *Integrated Windows Logon* section in the ***Administration Guide*** for details of setting up Integrated Windows Logon.

In addition to the procedures in the MyID documentation, you must also set up the authentication in IIS.

A PowerShell script called `ConfigureWindowsAuthentication.ps1` has been provided; this is installed on the MyID web server in the `Utilities` folder.

The script takes the following optional parameters:

- `webSiteName` – This is the name of the website that is hosting the MyID web service. By default, this is:

  ```
  Default Web Site
  ```

- `installationPath` – This is the folder where MyID was installed. By default, this is:

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

  If you do not specify this parameter, the script reads the installation folder from the registry.

The script ensures that Anonymous Authentication is set for MyIDDataSource and MyIDProcessDriver, and that Windows Authentication is enabled for the `WindowsAuth.asmx` web service.

Note: If you upgrade your MyID web services, you may have to run this PowerShell script again.

2.4 Configuring the MyID web services for 2-way SSL/TLS

See the *Two-way SSL/TLS* section in the ***Installation and Configuration Guide***.

3 Configuring the web services

You can carry out customization of the MyID Data Source and MyID Process Driver web services by editing files in the web service folders on the web server.

You can customize the following:

- The text of the Terms and Conditions to be accepted by cardholders when collecting a device.
- The on-screen text used for each part of the user interface in the client. This allows you to change the terminology used for individual elements or to translate the entire user interface on the client into another language.
- Filtering the jobs displayed to the user for specific device types, job types and job statuses for each client application. For example, you could set the Self-Service Kiosk to process only Activation jobs, while the Self-Service App was allowed to handle all other job types.
- Customize how MyID identifies the target user.
- Specify the certificate recovery web page for collecting soft certificates to an iOS device.
- Configure MyID to allow self-service operations.
- Set up a system to check the status of the web service.

3.1 Terms and conditions

Warning: Always back up your system before making any changes to the files in the web service folders.

The `TermsConditions.txt` file is located in the following folder by default:

```
C:\Program Files\Intercede\MyID\SSP\MyIDProcessDriver\Content\
```

These terms and conditions are displayed to a cardholder, who must agree to the conditions before being allowed to collect their device.

You can use a text editor to change the wording of this agreement.

After you have edited and saved the text file, recycle the **MyIDWebService** application pool in IIS to ensure that the web service is using the latest version of the file.

Note: Not all systems in MyID use this method to customize the terms and conditions. For more information about terms and conditions, see the *Customizing terms and conditions* section in the ***Administration Guide***.

3.2 Translating the client application user interfaces

For information about translating the text for all on-screen elements in the client applications, contact Intercede customer support, quoting reference SUP-138.

3.3 Job filtering

Warning: Always back up your system before making any changes to the files in the web service folders.

The `GetJobsRestriction.xml` file is located in the following folder by default:

```
C:\Program Files\Intercede\MyID\SSP\MyIDDataSource\Content\
```

You may not want every client application to handle every job that is available for the cardholder. For example, you may want your Self-Service Kiosks to handle only activation jobs, and require your cardholders to use their Self-Service Apps to handle all other jobs on their own workstations.

To do this, you can set up the `GetJobsRestriction.xml` configuration file to specify some or all of the following for each application:

- Job actions – you can specify that only jobs for particular actions are presented to the cardholder.
- Job statuses – you can specify that only jobs at specific statuses are presented to the cardholder.
- Enforced devices – you can specify that only jobs for selected device types are presented to the cardholder.
- Excluded devices – you can specify that jobs for selected devices are hidden from the cardholder.

Note: You cannot create a filter for enforced or excluded device types for jobs if the device is not known when the request is made; for example, for device issuance where the device has not been assigned at the request stage.

After you have edited and saved the XML file, recycle the **MyIDWebService** application pool in IIS to ensure that the web service is using the latest version of the file.

3.3.1 Job filtering configuration file format

The `GetJobsRestriction.xml` file is in XML format, with the following elements:

- `JobRestrictions`

  The top level element containing all of the content.

- `JobRestrictions/Platform`

  The element contains all of the restrictions for a specified client application.

  For example:

  ```
  <Platform id="4">
  ```

  This example identifies the Self-Service Application, which has an application code of `4`.

  Note: If the code for the client application accessing the web services does not appear in this configuration file, the web services will present an unfiltered list of jobs for all actions, all statuses, and all devices.

- `JobRestrictions/Platform/JobActions`

  If present, contains one or more `JobAction` elements that list the job actions that will be presented to the cardholder.

  If this element is not present, or does not contain any `JobAction` elements, all job actions will be presented to the cardholder.

- `JobRestrictions/Platform/JobActions/JobAction`

  Contains the name of a job action that will be presented to the cardholder.

  May contain an optional parameter `groupby` that allows you to set whether jobs are grouped together and presented to the user as a single job; for example, all jobs for a single device.

  For example:

  ```
  <JobAction>Activate</JobAction>
  ```

  ```
  <JobAction groupby="device">CardCertRenewal</JobAction>
  ```

  See section *3.3.2*, *Job actions* for a list of available actions.

- `JobRestrictions/Platform/Statuses`

  If present, contains one or more `Status` elements that list the job statuses that will be presented to the cardholder.

  If this element is not present, or does not contain any `Status` elements, all job actions will be presented to the cardholder.

- `JobRestrictions/Platform/Statuses/Status`

  Contains the name of a job status that will be presented to the cardholder.

  For example:

  ```
  <Status>Awaiting Issue</Status>
  ```

  See section *3.3.3*, *Job statuses* for a list of possible statuses.

- `JobRestrictions/Platform/EnforceDevices`

  If present, contains one or more `Device` elements that list the device types that will be presented to the cardholder.

  If this element is not present or does not contain any `Device` elements, *and* the `ExcludeDevices` element is empty or does not contain any `Device` elements, jobs for all device types will be presented to the cardholder.

- `JobRestrictions/Platform/EnforceDevices/Device`

  Contains the name of a device type for which jobs will be presented to the cardholder.

  For example:

  ```
  <Device>Oberthur PIV</Device>
  ```

- `JobRestrictions/Platform/ExcludeDevices`

  If present, contains one or more `Device` elements that list the device types that will not be presented to the cardholder.

  If this element is not present or does not contain any `Device` elements, *and* the `EnforceDevices` element is empty or does not contain any `Device` elements, jobs for all device types will be presented to the cardholder.

- `JobRestrictions/Platform/ExcludeDevices/Device`

  Contains the name of a device type for which jobs will not be presented to the cardholder.

  For example:

  ```
  <Device>BlackBerry</Device>
  ```

- `JobRestrictions/Platform/ExcludeCapabilities`

  If present, contains one or more `Capability` elements that list the credential profile capabilities that will not be presented to the cardholder.

  If this element is not present or does not contain any `Capability` elements, *and* the `EnforceCapabilities` element is empty or does not contain any `Capability` elements, jobs for all credential profiles will be presented to the cardholder.

- `JobRestrictions/Platform/ExcludeCapabilities/Capability`

  Contains the name of a credential profile capability for which jobs will not be presented to the cardholder.

  For example, you can exclude jobs for credential profiles that support Mobile devices:

  ```
  <Capability>Mobile</Capability>
  ```

  Exclusion of Mobile jobs is provided as default for the Self-Service App.

  See section *3.3.4*, *Capabilities* for a list of available capabilities.

- `JobRestrictions/Platform/EnforceCapabilities`

  If present, contains one or more `Capability` elements that list the credential profile capabilities that must be present in the credential profile for the job to be presented to the cardholder.

  If this element is not present or does not contain any `Capability` elements, *and* the `ExcludeCapabilities` element is empty or does not contain any `Capability` elements, jobs for all credential profiles will be presented to the cardholder.

- `JobRestrictions/Platform/EnforceCapabilities/Capability`

  Contains the name of a capability which must be present in the credential profile for jobs to be presented to the cardholder.

  For example, you can require that credential profiles must support Contact devices:

  ```
  <Capability>Contact</Capability>
  ```

  See section *3.3.4*, *Capabilities* for a list of available capabilities.

3.3.2 Job actions

Note: The job actions available may depend on your version and edition of MyID. For example, job actions that require activation are available in MyID PIV but not MyID Enterprise.

The following job actions are available:

- `Activate` – a card activation job for self-collection.
- `CardCertRenewal` – a card is being updated using a job that contains certificates that are being renewed.
- `CardIssuance` – a card is being issued using the standard procedure.
- `CardProfileChange` – a card is being updated to a different profile, as opposed to a new version of an existing profile.
- `CardReinstateJob` – a card is being reinstated.
- `CardReissue` – a card is being reissued.
- `CardReplacementIssuance` – a permanent replacement card is being issued.

- `CardReprovision` – a job which will re-issue the card with the latest content.
- `CardResync` – a card is being resynchronized.
- `CardTempReplacementIssuance` – a temporary replacement card is being issued.
- `LockPin` – a job which will lock the user PIN for the card.
- `ResetUnlockCode` – a job which will generate a reset unlock code for the card.
- `vsc_LockPin` – a job which will lock the user PIN for the virtual smart card.
- `vsc_ResetUnlockCode` – a job which will generate a reset unlock code for the virtual smart card.

The actions supported depend on the client application you are using:

Job Action	SSA	SSK	Android	iOS	SSA Automation
Activate	✓	✓			
CardCertRenewal	✓	✓	✓	✓	
CardIssuance	✓	✓			
CardProfileChange	✓	✓	✓	✓	
CardReinstateJob	✓	✓	✓	✓	
CardReissue	✓	✓			
CardReplacementIssuance	✓	✓			
CardReprovision			✓	✓	
CardResync	✓	✓	✓	✓	
CardTempReplacementIssuance	✓	✓			
LockPin					✓
ResetUnlockCode					✓
vsc_LockPin					✓
vsc_ResetUnlockCode					✓

3.3.3 Job statuses

The following job statuses are available:

- `Awaiting Issue`
- `Awaiting Validation`
- `Awaiting Encoding`
- `Awaiting Activation`
- `Completed`

Note: Not all job statuses are appropriate for self service clients. Contact customer support for more information.

3.3.4 Capabilities

The following capabilities are available:

- `Contact` – smart cards with contact chips.
- `Contactless` – smart cards with contactless features.
- `MVSC` – Microsoft Virtual Smart Cards. Your installation of MyID may require an update to support VSCs.
- `Physical` – cards with only a magnetic stripe.
- `SoftCert` – soft certificates.
- `Mobile` – mobile credentials.
- `OTP` – one-time password tokens.

3.3.5 Example job filtering configuration file

```
<?xml version="1.0" encoding="utf-8" ?>
<JobRestrictions>
  <!-- Self Service app (4) -->
  <Platform id="4">
    <!-- Opt in to these job types-->
    <JobActions>
      <JobAction>Activate</JobAction>
      <JobAction>CardResync</JobAction>
      <JobAction>CardReissue</JobAction>
      <JobAction>CardProfileChange</JobAction>
      <JobAction>CardReinstateJob</JobAction>
      <JobAction>CardIssuance</JobAction>
      <JobAction>CardReplacementIssuance</JobAction>
      <JobAction>CardTempReplacementIssuance</JobAction>
      <JobAction groupby="device">CardCertRenewal</JobAction>
    </JobActions>
    <Statuses>
      <Status>Awaiting Issue</Status>
    </Statuses>
    <!--Enforce Devices-->
    <!--<EnforceDevices>
      <Device>Oberthur PIV</Device>
    </EnforceDevices>-->
    <!-- Opt out of these device types -->
    <ExcludeDevices>
      <Device>BlackBerry</Device>
    </ExcludeDevices>
    <!-- Exclude Jobs when these capabilities are present in the card profile -->
    <ExcludeCapabilities>
      <Capability>Mobile</Capability>
    </ExcludeCapabilities>
    <!-- Enforce that these capabilities are present in the card profile -->
    <!-- <EnforceCapabilities>
      <Capability>Contact</Capability>
    </EnforceCapabilities> -->
  </Platform>
</JobRestrictions>
```

3.4 Specifying the target user

The user identifier passed to the MyID server is based on the Windows logon name of the user. This is then matched against the SAM Account Name stored for the user in the MyID database.

You can change how the system handles the user identifier in the following ways:

- Set the `/un` option on the command line of the Self-Service App to the logon name you want to use.
- To change the identifier that is passed to the web services, set the Windows environment variable `MYID_USERNAME` to the identifier you want to use. This value is used instead of the Windows logon name for all users on the PC.

  Note: This environment variable has no effect if you launch the Self-Service App using a hyperlink. To specify a different logon name, you must use the `/un` command line option instead.
- To change which MyID field the identifier is matched against, alter the `ws_LogonJobs` view in the MyID database to change the definition of the `UserIdentifier` field to point to a different field. This allows you to compare the user identifier to a field other than the SAM Account Name for the user.

  Note: Any installation of a MyID update may affect the `ws_LogonJobs` view in the MyID database; after you update MyID, you must check the `ws_LogonJobs` view in the database and, if necessary, re-apply any customizations.

Note: In addition to the Windows logon name, MyID also passes the User Principal Name from the client and attempts to match this against the UPN stored for the user in the MyID database; however, if you use the `/un` command line option or the `MYID_USERNAME` environment variable to override the Windows logon name taken from the client, MyID does *not* pass the User Principal Name from the client.

3.4.1 Case sensitivity

- When MyID matches the User Principal Name from the client against the UPN stored in the database, it carries out a case-insensitive match.
- When MyID matches the Windows logon name against the SAM Account Name stored in the database, it carries out a case-insensitive match.
- When MyID matches the username provided by the `/un` command line option or the `MYID_USERNAME` environment variable against the SAM Account Name stored in the database, it carries out a case-insensitive match.

3.4.2 Disabling UPN and SAMAccountName checks for the Self-Service App

If you launch the Self-Service App without the `/un` parameter and there is no `MYID_USERNAME` environment variable configured, by default, MyID carries out a series of checks, including attempts to find:

- The **User Principal Name** (UPN) obtained by the client, which is the `UserPrincipalName` in the database.
- The **SAM Account Name** obtained by the client, which is the `SAMAccountName` in the database.

If you do not have a UPN or SAM Account Name, these checks fail, and you cannot view your jobs in the Self-Service App.

To remedy this, you can set the **Ignore UPN and SAMAccountName checks for Self-Service jobs** configuration option (on the **Self-Service** page of the **Security Settings** workflow) to `Yes`, and MyID ignores the UPN and SAM Account Name checks, allowing you to view your available jobs.

3.5 Certificate recovery web page

When recovering certificates to an iOS device using the **Collect My Soft Certificates** workflow, the web services use an intermediate web page to present a link to the PFX files. The app loads the pages in the Safari browser and the user selects the link to download the PFX files.

Note: By default, when MyID issues software certificates, it encrypts the passwords protecting the PFX files using AES256/SHA2. However, some Operating Systems do not support this modern security standard, which creates a problem when importing the certificates onto these; for example, any Apple OS (macOS or iOS), any Windows Server OS lower than Windows 2019, and any Windows client OS lower than Windows 10 build 1709. If you want to import software certificates onto an OS that does support not the encryption of PFX files using AES256/SHA2, you must set the **Use SHA1 encryption for certificates issued as PFX files** option in the **Server** tab of the **Security Settings** workflow to `Yes`.

To present the PFX files to the user, the certificates are converted into an XML file that is transformed into HTML using XSL. If required, you can modify the transform file to present the PFX files to the user.

The transform file is `PFX-512-Download.xslt`, and is installed to the following folder by default:

```
C:\Program Files\Intercede\MyID\SSP\MyIDProcessDriver\Transforms\
```

Note: If you provide any images in your transform, you are recommended to use absolute paths rather than relative paths.

The standard transform file displays a simple HTML page with a link to the PFX files that are provided in the `/Certificates/certificate/PFXFileName` nodes of the XML. The readable name in `/Certificates/certificate/CertPolicy` is used for the text of the link.

3.5.1 Available attributes

The XML comprises a top-level `Certificates` node containing one or more `certificate` nodes. Each `certificate` node contains the following attributes.

Note: Not all attributes are relevant to soft certificates.

- `ID` – The ID of the certificate.
- `LogonName` – The logon name of the certificate owner.
- `DeviceSerialNo` – The serial number of the device. For example, `Certificate Package 51344` for a soft certificate package.
- `DeviceTypeName` – The type of device. For example, `System Certificates` for a soft certificate.
- `CertSerialNo` – The serial number for the certificate.
- `CertStatus` – The MyID status code for the certificate.
- `CertTemplate` – The CA template or policy used to issue the certificate.
- `Collected` – The ID of the collected status. Maps to the `ID` column of the `Collected` table in the MyID database.
- `ContainerName` – The name of the container for the certificate. For example, `FILE` for a soft certificate.
- `CertPolicy` – The readable name of the certificate policy used to issue the certificate.
- `KeyArchived` – The ID of the archive status of the certificate:
  - `0` – Not archived.
  - `1` – Archived on the CA.
  - `2` – Archived in MyID.
- `DatetimeStamp` – The time the certificate was added to the MyID database.
- `RevocationCode` – Not applicable.
- `RevokeComment` – Not applicable.
- `ErrorText` – Not applicable.
- `DeleteContainers` – Always 0 for soft certificates.
- `PKCS12` – Not applicable.
- `PKCS7` – A hex-encoded PKCS#7 certificate.
- `PathToCer` – Not applicable.
- `PathToPFX` – The path to the PFX file containing the certificate.
- `PFXFileName` – the name of the PFX file containing the certificate, without the path. The user must click on a link to this file to install the certificate.
- `BasePath` – Not applicable.
- `RelativePath` – Not applicable.
- `VerifiedExternally` – Not applicable.

3.6 iOS OTA web page

When using iOS OTA to issue certificates to an iOS device, the web service uses an intermediate web page to present a link to the CA root certificate and the Enroll page used to provision the certificates.

The web page is generated by transforming XML into HTML using XSL. If required, you can modify the transform file.

The transform file is `ScepProvision.xslt`, and is installed to the following folder by default:

```
C:\Program Files\Intercede\MyID\SSP\MyIDProcessDriver\Transforms
```

Note: If you provide any images in your transform, you are recommended to use absolute paths rather than relative paths.

Note: By default, `ScepProvision.xslt` file contains a meta refresh node which automatically takes the user to the `EnrollUrl`. This can be removed or the time taken (defaults to 0 seconds) can be changed if required.

3.6.1 Available attributes

The XML comprises a top-level `Parameters` node containing the following elements:

- `CaUrl` – The URL which can be used to download the root CA certificate. This is optional and will not be required if all devices are preconfigured to trust the root CA certificate.
- `EnrollUrl` – The URL which needs to be followed to begin the process of issuing the certificates.

  Note: You must either include a hyperlink to the `EnrollUrl`, or a meta refresh node that automatically takes the user to the `EnrollUrl`.

In addition to the `EnrollUrl` being mandatory, a link with:

```
href="myidmcm://completed"
```

is also mandatory so that the user can be returned to the Identity Agent application to complete the enrollment.

3.7 Security for self-service operations

MyID has implemented a series of security features where, amongst other security considerations, it is no longer possible to determine a username from just a serial number. This limitation prevents some self-service operations; for example, **Unlock My Mobile** on the mobile platforms.

The issue may present with an error similar to:

```
This logon mechanism isn't available with the current configuration.
501107
```

To configure MyID to allow the previous behavior, edit the `myid.config` file in the `MyIDProcessDriver` folder. Set the value of the key `PreventStartWorkflowWithPassphraseByDevice` to `false` to disable this feature.

3.8 Checking the status of the web services

You can use the `IsAlive` API method on the web service to confirm that the web services are running and reachable. For example, you may want to check that the web services are running before launching the Self-Service App.

To check the status, call the following method:

```
https://<server>/MyIDProcessDriver/ProcessDriver.asmx/IsAlive
```

where:

- `<server>` is the server name of your web services server.

This method returns the Boolean value `true` if the web services are running; for example:

```
<boolean xmlns="https://www.intercede.com/myid">true</boolean>
```

3.9 Reverse proxies and load balancing

If you have a reverse proxy in front of the MyID web services servers, for example for load balancing, you may have to carry out additional configuration.

If, in MyID Desktop, you can access some workflows (for example, **Collect Card** or **Erase Card**) but not others (for example, **Edit Person**) this can be caused by the reverse proxy. By default, the MyID web services use the requesting path to generate various other paths that are passed back to the client; as the reverse proxy has changed this path, the generated paths returned to the client are not correct.

To address this, you can provide fixed URLs for the paths in the web service configuration file:

1. Back up the `myid.config` configuration file.

   On the web services server, this is located in the following folder by default:

   ```
   C:\Program Files\Intercede\MyID\SSP\MyIDProcessDriver\
   ```

2. Open the `myid.config` file in a text editor.

3. Locate the following lines:

   ```
   <add key="MyIDSessionUrl" value="
   {0}/myid/default.asp?dest=/timeout.asp?action=ping&amp;lang=[lang]"/>
   <add key="AuthenticationUrl" value="
   {0}/myid/default.asp?dest=/hyperoptionInFrame.asp?passthroughauthenticati
   on=true&amp;lang=[lang]"/>
   <add key="WebProcessUrl" value="
   {0}/myid/default.asp?dest=/hyperoptionInFrame.asp?option=
   {2}&amp;hideMenuBar=true&amp;backLink=desktopDone.asp&amp;lang=[lang]"/>
   <add key="AbortUrl" value="
   {0}/myid/default.asp?dest=/CompleteTask.asp?Status=Abort&amp;lang=
   [lang]"/>
   <add key="EndWorkflowUrl" value="
   {0}/myid/default.asp?dest=/EndSession.asp&amp;lang=[lang]"/>
   ```

4. Replace the `{0}` substitution token in each of the above lines with the protocol and server address; for example:

   ```
   https://myserver.domain.com
   ```

   The edited lines will now be similar to the following:

```
<add key="MyIDSessionUrl" value
="https://myserver.domain.com/myid/default.asp?dest=/timeout.asp?action=p
ing&amp;lang=[lang]"/>
<add key="AuthenticationUrl" value
="https://myserver.domain.com/myid/default.asp?dest=/hyperoptionInFrame.a
sp?passthroughauthentication=true&amp;lang=[lang]"/>
<add key="WebProcessUrl" value
="https://myserver.domain.com/myid/default.asp?dest=/hyperoptionInFrame.a
sp?option={2}&amp;hideMenuBar=true&amp;backLink=desktopDone.asp&amp;lang=
[lang]"/>
<add key="AbortUrl" value
="https://myserver.domain.com/myid/default.asp?dest=/CompleteTask.asp?Sta
tus=Abort&amp;lang=[lang]"/>
<add
key="EndWorkflowUrl" value
="https://myserver.domain.com/myid/default.asp?dest=/EndSession.asp&amp;
lang=[lang]"/>
```

Note: Do not do a global search and replace in the configuration file. The `{0}` substitution token is used in other configuration options for other purposes.

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

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

Note: The `myid.config` configuration file is a core MyID file that may be overwritten when you update or upgrade MyID. You must implement your changes again after updating or upgrading. Make sure you use the latest version of the configuration option as your basis for the substitution; in particular, MyID 12.8 introduces the use of `dest=/EndSession.asp` instead of `dest=/blank.html` in the configuration file, which is essential for signing out all aspects of the session when signing out from the MyID Operator Client.