intercede

MyID CMS

MyID Enterprise
Version 12.10

Advanced Configuration Guide

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 contains information on configuring your MyID® system, including security considerations, monitoring, using archive databases, setting up email, and other advanced configuration.

Make sure you have followed the *Installation and Configuration Guide* to install and set up your system for basic operation before setting up any of the advanced options.

2 Securing the MyID application

> **Important:** For your production environment, you *must* ensure that your MyID system is secure. The latest ***System Security Checklist*** document provided with MyID provides important information about risks and recommendations for a wide variety of security considerations, and you are strongly advised to complete the checklist for your system.

The following sections include additional information about securing your system – whether you choose to implement them depends on the security requirements of your organization. Note, however, that by default MyID requires some security options that you must explicitly disable if you do not want to use them; see section *2.1*, *Device security settings* for more information.

Note: You should carry out the additional lockdown procedures after you have installed MyID. You may experience problems if you attempt to install MyID on a system that has already been locked down.

2.1 Device security settings

When you install MyID, the settings on the **Device Security** page of the **Security Settings** workflow are configured to require you to use customer GlobalPlatform keys and random Security Officer PINs (SOPINs). The system is also configured to display warnings if your system is not securely configured; for example:

```
The system is not configured for production use – check the MyID system
security checklist document for further information.
```

If this warning appears, you must review the settings on the **Device Security** tab on the **Security Settings** workflow; see the ***System Security Checklist*** document. This document also contains information about configuring SOPINs, GlobalPlatform keys, and PIV9B keys to ensure that your system is secure and configured for production use.

2.2 Access to the MyID web application

The installation program sets up the MyID website to use anonymous authentication using the IIS user you specified when installing MyID. You can configure IIS to provide further security if required.

2.3 Secure access to diagnostic files

There are various diagnostic files that exist within MyID to help Intercede determine where issues might lie in the system. These files may sometimes return information that is considered private.

These files are located in the `diagnostics` subfolder within each language folder on the MyID web server. This subfolder is not accessible anonymously.

By default, this subfolder is locked to prevent *any* access. If you need to access the diagnostic features, including Automated Testing, contact customer support, quoting reference SUP-101.

2.4 MyID and SQL Server permissions

SIU references: SIU-127, SIU-131, SIU-318, SIU-319, SIU-320.

> **Warning:** If running MyID with a named user, make sure the MyID COM+ account is added with 'English' as the default language, or date formats will cause failures.

The account used for database access (the MyID COM+ account) is assigned the permissions needed to create and use the MyID databases when MyID is installed. If you want to reduce the level of these permissions following installation, you must ensure that the account being used keeps the following levels of access as a minimum.

- The account *must* have the following roles on the MyID databases:

  - `public`

  - `db_datareader`

  - `db_datawriter`
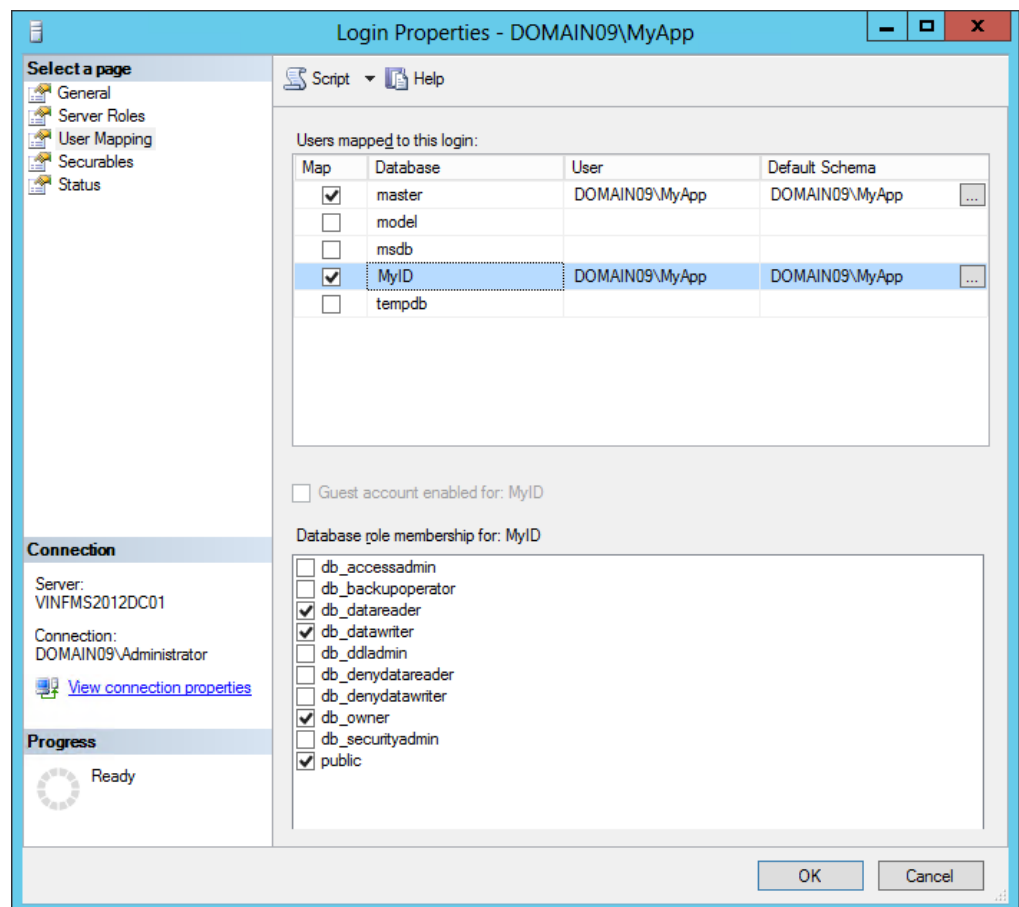
Set these permissions in SQL Server Manager. Under the database instance, select **Security > Logins**, then right-click the MyID COM user.

- Ensure that the **Default Schema** is set to `dbo` or another appropriate setting; a default schema of `sys` will cause connection problems.

- The stored procedures executed by the system also need to allow execute permission to the MyID COM user. This includes all 'User' type stored procedures in the MyID database. You can assign permissions to stored procedures individually, or grant `db_owner` access to the MyID COM user in **Security > Logins**.

- Authentication

You specify whether to use SQL Server authentication or Windows authentication when installing MyID.

- For Windows Authentication to operate, the MyID application server must belong either to the same domain as the database server or to a trusted domain.

2.5   MyID and COM+ permissions

SIU references: SIU-154, SIU-155, SIU-156, SIU-157, SIU-158, SIU-159, SIU-160, SIU-161, SIU-162, SIU-163, SIU-164, SIU-165, SIU-166, SIU-167, SIU-168, SIU-169, SIU-170, SIU-171, SIU-172.

These permissions are set automatically during the installation process.

If you want to review or change these permission settings, they can be found in the **Properties** of the specified COM+ DLL. In **Component Services > My Computer > COM+ Applications**, right click the component then select **Properties**.
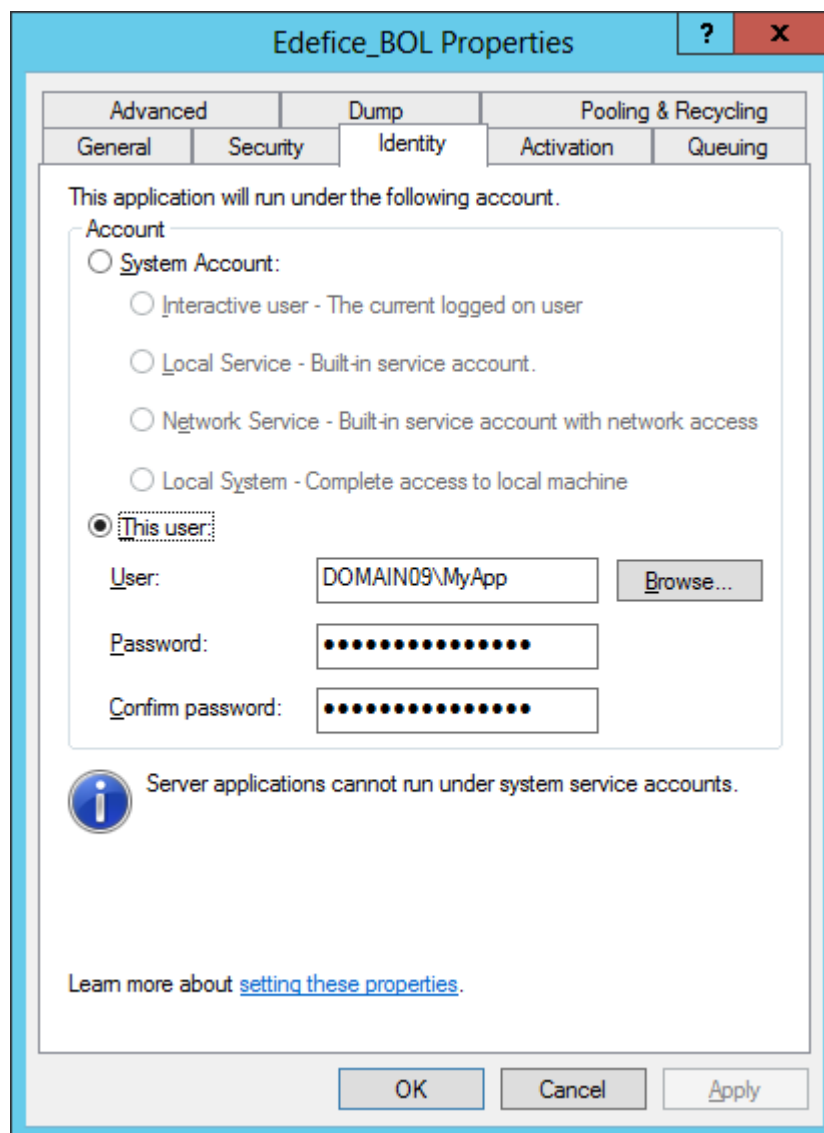
MyID installs some or all of the following COM objects, depending on the options you selected during installation:

- APDUCardServer
- EAudit
- eCS
- Edefice_BOL
- Edefice_CS
- Edefice_DAL
- eEventLog
- eExternalDataSource
- Entrust_Admin
- ePKIConfig
- ImportProcessor

2.6   MyID startup

If you are using an HSM that requires password entry, you can use the Card Manager Startup utility on the application server to enter your HSM credentials and control the operation of the MyID eKeyServer service that secures the MyID application.

Note: You must set the Startup utility to run when the MyID application server starts up.

2.6.1   Using the Startup utility

To start the utility:

1. Open the Windows Start menu.

2. In the MyID group, select **Startup**.

3. Locate the **Startup** icon in the Windows **Start** menu, right-click and select **Run as Administrator**.

    The Startup utility now starts. The logged-on user must have permission to access the MyID database and stop or start the eKeyServer service.

You can also set the Startup utility to run as administrator when the system starts up. See your Windows documentation for details. For example, create a shortcut to the utility in:

```
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp
```

Right-click the shortcut, select **Properties**, then on the **Compatibility** tab select **Run this program as an administrator**.

2.6.2   Using Startup with an HSM-based master key

When you log on to Windows on the server, the MyID **Startup** utility should run automatically. If it does not, select **Startup** from the **MyID** folder of the **Start** menu.

1. The **Startup** box is displayed.
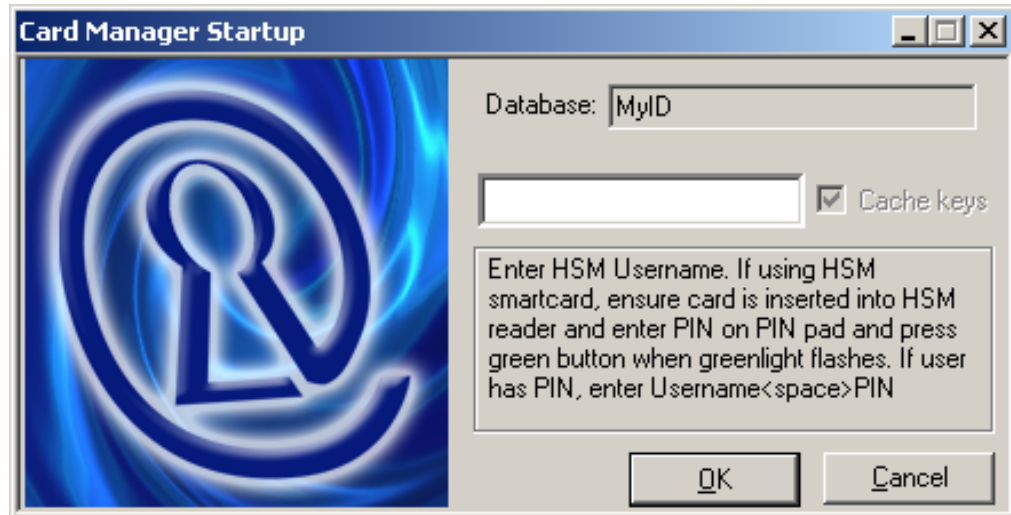
If the eKeyServer service is already running, a warning message is displayed and only the **Cancel** button is active.

2. Depending on the type of user created on the HSM, you must do one of the following:

   - If the user was created with **No Authentication**, type the username.

   - If the user was created with **PIN Authentication**, type the username, then a space, then the PIN.

   - If the user was created with **Smartcard Authentication**:

      a. Type the username.

      b. Insert the smart card into the HSM card reader.

      c. Enter the PIN on the PIN-Pad on the HSM card reader when the green light flashes.

   Note: You cannot use the Cache keys option for an HSM-based master key. If you want to set up unattended startup, you must configure this when you first run GenMaster, if your HSM supports it.

3. The Startup utility then starts the eKeyServer service, which takes a few seconds.

   The message "Key Server is now active" is displayed when the service has successfully started.

4. Click **Close**.

5. The system tray shows the status of the eKeyServer service. A gray icon indicates that the service is not running. Right-click the icon to display the pop-up menu:

   - **Start** to start the eKeyServer service.

   - **Stop** to stop the eKeyServer service.

- **Restart** to stop the eKeyServer service and then restart.

- **Exit** to exit the Startup application, leaving the eKeyServer service running.

2.6.3   Startup utility procedure

The Card Manager Startup utility requires the appropriate permissions to carry out its processes. The utility carries out the following steps – you must make sure that your users are configured with the correct permissions for each step.

1. The utility reads the registry.

    For the `Mastercard` key in the registry, and any subkey beneath it:

    - The MyID COM+ user needs read access.

    - Any user that logs onto the application server to use the utility needs read access.

    - In the rare occasions where the `Mastercard` part of the registry needs to be changed by MyID, full control to the `Mastercard` key and its subkeys is required for the MyID COM+ user account, and the logged on user. Situations that require this part of the registry to be updated include:

        - Running GenMaster for the first time.

        - Using the cache keys feature where master cards have previously been used.

    The `Mastercard` is located in the following part of the registry on the application server:

    `HKEY_LOCAL_MACHINE\SOFTWARE\Intercede\Edefice\MasterCard`

2. Launches the Edefice_DAL component and pull back configuration information from the database.

    This is launched by the utility using the logged-on user account's permissions.

3. Attempts to start the eKeyServer service.

    Again, this uses the permissions of the logged-on user account.

2.7   **eKeyServer Service**

The eKeyServer service is automatically configured to run using the MyID COM+ account, specified during the installation of MyID server.

If you need to change this account for any reason, you must ensure that the new account has `Read` rights to the `.udl` files in the `System32` folder that are used to access the database server. No additional permissions need to be assigned to this account.

The format of the user name must be `DOMAIN\UserName` rather than the form returned by the **Browse** feature on the property page (`UserName@DOMAIN`).

If the account selected does not already have **Log On As A Service** rights on the MyID server, the Service Control Manager MMC snap-in automatically assigns those rights and displays a confirmation message.

To verify the status of the eKeyServer service, open the **Service Control Manager**, (by selecting **Administrative Tools** then **Services**) from the control panel.
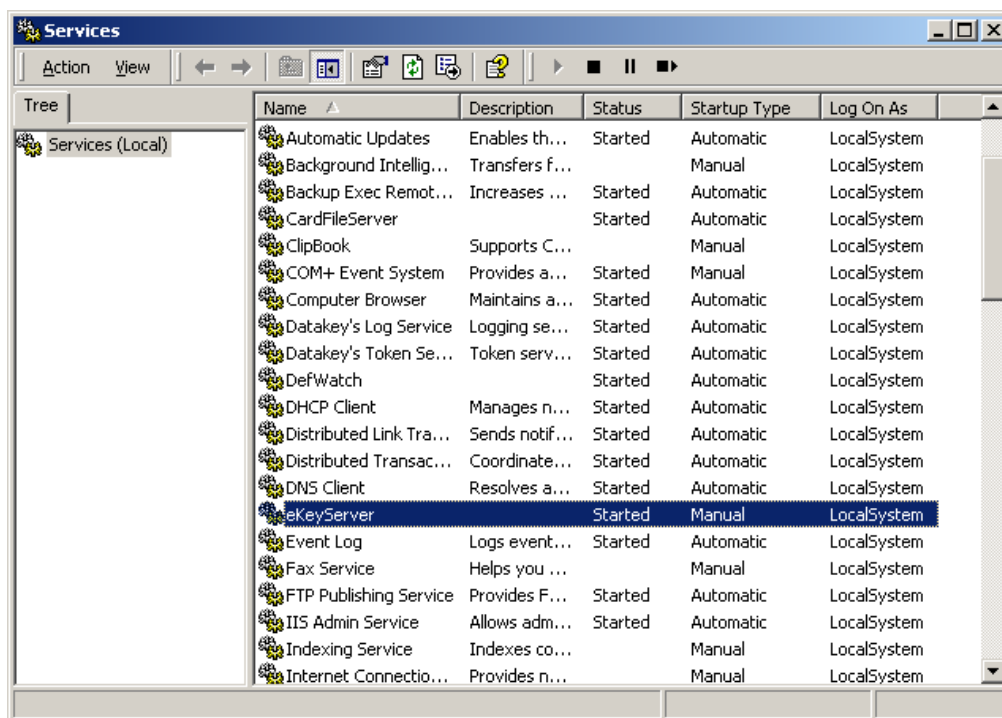
2.8   Protecting the registry

Warning: If eKeyServer is running in 'cached key' mode, the entries in the registry *must* be protected.

You can protect the registry using `regedt32.exe` to set permissions on the `HKEY_LOCAL_MACHINE\SOFTWARE\Intercede\Edefice\MasterCard` branch of the registry on the application server. Read permission is needed for normal use, write permission is required when creating master keys that stored in the registry.

Make sure the following users have read access to this branch of the registry:

- The MyID COM user.

- Any user who logs on to the application server to use the Startup utility.

Make sure the following users have full control over this branch of the registry and its subkeys when running GenMaster for the first time:

- The MyID COM user.

- The logged-on user.

3 Monitoring MyID

MyID provides several methods that allow you to monitor the status of your system.

- Expiry of system credentials.

  MyID keeps track of the expiry dates of the Windows user accounts used to run the MyID components and services, and the certificates used to secure the MyID services, sign data objects, or authenticate to other systems. The status and expiry date of each of these items is listed on the **System Credentials** page of the **System Status** report.

- System health check.

  MyID provides a web service you can use to perform internal diagnostics and report the result. These diagnostics include performing a number of checks to check the integrity of the installation, then inspecting every method that can be implemented through DCOM to the MyID application server to ensure that the method is available.
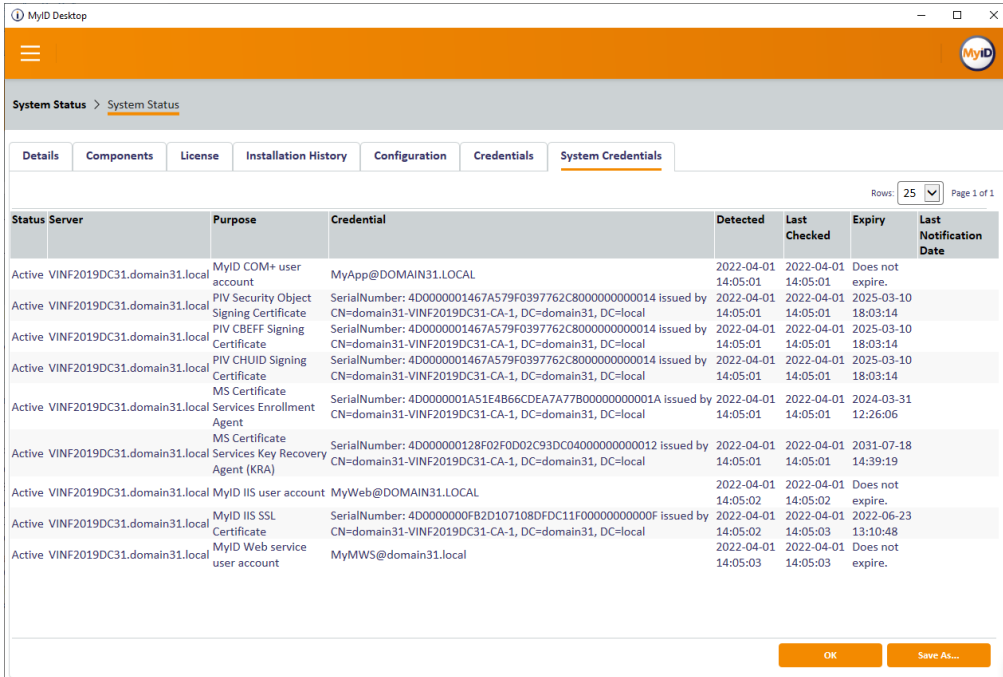
  MyID also provides a simple check of the status of the MyID web services server even if you have not enabled the health check service. See the *Checking the status of the web services* section in the ***Web Service Architecture*** guide.

- Connectivity.

  MyID supports sending both email and SNMP trap notifications from the server, and provides its own SNMP agent to allow external software to monitor the status of MyID's systems. This allows you to monitor such cases as a key server failure to connect to a HSM, database access layer connection failures, or certificate authority failures.

3.1 Monitoring the expiry of system credentials

MyID keeps track of the expiry dates of the Windows user accounts used to run the MyID components and services, the certificates used to secure the MyID services, and the user account used to access the SMTP server. The system tracks the expiry of the account passwords and the accounts themselves. The status and expiry date of each of these items is listed on the **System Credentials** page of the **System Status** report.

Note: When you first install MyID, this table will be blank. The expiring item features runs to a schedule (by default, every day at 0300 UTC – see section *3.1.5*, *Changing the schedule*) – this table is populated the first time the system checks the expiry of the accounts and certificates.

The **Status** column displays **Active** for all credential or certificates that are being monitored, and **Superseded** for any credential or certificate that has been replaced; for example, if you renew the email signing certificate, it creates a new entry for the new certificate, and lists the old certificate with a status of **Superseded**.

MyID also sends notification email messages for expiring and expired user accounts and certificates.

The following email templates are used:

- `ServicePasswordExpiring` – sent 28, 21, 14, 7, 3, 2, and 1 days before a user account expires.

- `ServicePasswordExpired` – sent when a user account has expired.

- `CertificateExpiring` – sent 90, 60, 28, 21, 14, 7, 3, and 2 days before a certificate expires.

- `CertificateExpired` – sent when a certificate has expired.

- `CredentialExpiring` – sent 28, 21, 14, 7, 3, 2, and 1 days before the user account used

for the SMTP server expires.

- `CredentialExpired` – sent when the SMTP user account has expired.

You can use the **Email Templates** workflow to edit the content of these email messages. You must set up email in MyID to allow the system to send these messages – see section *5*, *Setting up email* for details.

3.1.1 The monitoring services

The services that monitor expiring items run on each MyID application server and web server. There is one service for each user account, with the following names:

- **MyID Expiring Items: App** – used to monitor the MyID COM+ user account along with any certificates assigned to it.
- **MyID Expiring Items: Web** – used to monitor the MyID IIS user account along with any certificates assigned to it.
- **MyID Expiring Items: Mws** – used to monitor the MyID web services user account along with any certificates assigned to it.

Each service monitors its account and certificates, and does so whether it is in the same domain as the rest of the system or it is in a web server in a DMZ.

The **MyID Expiring Items: Web** and **MyID Expiring Items: Mws** services check SSL/TLS certificates on the servers on which they are running, and the iOS OTA signing certificate, if that feature is configured.

The **MyID Expiring Items: App** service checks all other certificates, and the user account used for SMTP server authentication.

3.1.2 Monitored system credentials

The system credential monitoring system checks the status of the following user accounts and certificates:

Item	Requirements	More information
MyID COM+ user account		See the *Setting up user accounts* section in the **Installation and Configuration Guide**.
MyID IIS user account		See the *Setting up user accounts* section in the **Installation and Configuration Guide**.
MyID web service user account		See the *Setting up user accounts* section in the **Installation and Configuration Guide**.
iOS OTA Signing Certificate	Available if your system is configured for iOS OTA.	See the *Setting up iOS OTA provisioning* section in the **Mobile Identity Management** document for details.
Mobile Signing Certificate	Available if your system is configured for mobile issuance.	See the *Setting the content signing certificate* section in the **Mobile Identity Management** document for details.

Item	Requirements	More information
SCEP Signing Certificate	Available if your system is configured for SCEP.	See the *Signing and encryption certificates for SCEP* section in the **Administration Guide**.
SCEP Encryption Certificate	Available if your system is configured for SCEP.	See the *Signing and encryption certificates for SCEP* section in the **Administration Guide**.
Email Signing Certificate	Available if you have configured email signing.	See section *5.1*, *Signing email messages* in this document.
MS Certificate Services Enrollment Agent	Available if you are using a Microsoft CA.	See the *Enrollment Agent certificate* section in the **Microsoft Windows CA Integration Guide**.
MS Certificate Services Key Recovery Agent	Available if you are using a Microsoft CA.	See the *Encryption key recovery* section in the **Microsoft Windows CA Integration Guide**.
Generic Certificate Authority	Available if you are using any supported certificate authority where MyID is configured to use a certificate to connect to the CA.	See the integration guide for your certificate authority.
MyID IIS Web Site Certificate	Available if you have configured the MyID website to use `https`.	See the *Configuring SSL/TLS (HTTPS)* section in the **Securing Websites and Web Services** guide.
Notification SSL Certificate	Available only on systems that have been customized to use SSL/TLS for notifications.	See the additional documentation provided with your customization.
Signing Certificate	Available if you have set up a CVC signing certificate for OPACITY.	See the *Setting up OPACITY* section in the **Smart Card Integration Guide**.
SMTP user	Used if your SMTP server requires authentication.	See section *5*, *Setting up email*.

3.1.3 Changing service account passwords

If you need to change the password for the MyID COM+, IIS, or web service user accounts, you can use the Password Change Tool. See the **Password Change Tool** guide for details.

3.1.4 Replacing expiring certificates

See the instructions in the MyID documentation that you followed to set up the certificate originally. Make sure you pay attention to any additional instructions for replacing certificates; for example, for the Enrollment Agent certificate on a Microsoft CA, you must move the old

certificate out of the certificate store before you can replace it.

3.1.5 Changing the schedule

By default, the monitoring service runs once a day at 0300 UTC. If you need to change this, you can edit the `ExpiringItemsServiceSchedule` table in the MyID database. Set the `StartTime` to the time in UTC that you want the service to run, using the format `HH:MM:SS`, and the `ThenEvery` field to the number of days between each run.

For example, to set the service to run once a week at 0630 UTC:

```
Update ExpiringItemsServiceSchedule set StartTime = '06:30:00', ThenEvery = 7
```

The services will pick up the new schedule the next time they run – that is, at the old start time. If you want the service to pick up the new schedule immediately, restart the services.

3.2 System health check

MyID provides a web service you can use to perform internal diagnostics and report the result. These diagnostics include performing a number of checks to check the integrity of the installation, then inspecting every method that can be implemented through DCOM to the MyID application server to ensure that the method is available.

3.2.1 Setting up the health check service

By default, the health check service is disabled. To enable the health check service:

1. On the web services server, open the `myid.config` file in a text editor.

    By default, this file is in the following folder:

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

2. In the `<MyIDSettings>` node, add the following:

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

3. Save the file.

Note: You can perform a simple check of the status of the MyID web services server even if you have not enabled the health check service. See the *Checking the status of the web services* section in the ***Web Service Architecture*** guide.

3.2.2 Viewing the report

To view the report, visit the following URL:

```
<server>/myidprocessdriver/healthcheck.aspx
```

Where `<server>` is the address of your MyID web services server. For example:

```
https://myserver/myidprocessdriver/healthcheck.aspx
```

By default, the report is displayed in a readable format:

To return the results as machine-readable XML or JSON, append one of the following to the URL:

- `?format=xml`

- `?formal=json`

For example:

```
https://myserver/myidprocessdriver/healthcheck.aspx?format=xml
```

```
https://myserver/myidprocessdriver/healthcheck.aspx?formal=json
```

3.2.3   Troubleshooting

3.2.3.1 COM+ errors

Errors in the Object Creation section of the report (marked in red, with the word Failed, and error details such as "Method does not exist") may be caused by:

- Problems communicating with the application server.

- Issues with the MyID COM+ user account – check that the account is valid, and has not changed its password. If you have changed the password, you can use the Password Change Tool to update MyID with the details. See the **Password Change Tool** guide for details.

- Issues with the COM+ proxies on the MyID web services server. Make sure all the COM proxies have been installed; see the *Split deployment* section in the **Installation and Configuration Guide** for details.

- Issues with incorrectly installed COM+ components.

If you have any issues, you can use the System Interrogation Utility to check that your system is configured correctly. See the **System Interrogation Utility** guide for details.

3.2.3.2 Database errors

If your database is not working correctly, you may see errors similar to the following:

- SQL error `2800` – this may be caused when your server is running, but the MyID database is offline.

- SQL error `HYT00` – this may be caused when the database server is not running at all.

If you experience database errors, check that your database is running correctly. To help troubleshoot your issues, you can use the **Test Connection** option on the Data Link Properties dialog – on the MyID application server, in the Windows `System32` folder, double-click the MyID `*.udl` file that connects to the database that is producing the errors. See your Microsoft documentation for details of any ODBC error codes that appear.

3.3 Monitoring connectivity

MyID supports sending SNMP trap notifications from the server, and provides its own SNMP agent to allow external software to monitor the status of MyID's systems.

By default, both methods are disabled. You must enable them before you can use them.

Note: MyID currently supports only SNMP v1 and v2c versions of the SNMP protocol.

If the monitoring system detects a problem, you can use the MyID **System Events** workflow to find additional information about the problem. For KeyServer or HSM issues, the Windows event log may provide additional details.

3.3.1 Supported monitoring

MyID supports the following monitoring methods:

MyID Component	Description	SNMP trap	Email	SNMP agent
`eKeySrv`	KeyServer used for encryption and decryption.	✓		✓
`Edefice_DAL`	Database access layer.			✓
`eCS`	Certificate authority access layer.	✓	✓	✓

3.3.2 SNMP attributes

Each component currently supports the same SNMP attributes, as defined in the `MyIDMIB` file.

Name	Description
Error ID	A unique, per component, value for the error.
Error Description	A short descriptive name for the associated error ID.
Error Message	The original error message associated with this error.

The `eCS` component has the following possible errors:

Error	Description
1	Unknown error
2	Certificate Authority unavailable

The `eKeySrv` component has the following possible error:

Error	Description
1	Unknown error

The `Edefice_DAL` component has the following possible errors:

Error	Description
1	Unknown error
2	Could not connect
3	Access violation

3.3.2.1 MIB definition file

MyID provides a `MyIDMIB.mib` file, which provides a standard MIB definition format of all the MIB objects that the MyID can report. Load this file into your external SNMP monitoring system so that it can identify the objects and attributes being reported.

The MIB file is installed to the `Utilities` folder on the MyID application server; by default, this is:

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

3.3.3 SNMP trap notifications

MyID provides the ability to send SNMP Trap notifications from the following systems:

- KeyServer – used to provide secure encryption and decryption. Used for communication with HSMs.
- Certificate Server – used to communicate with certificate authorities.

To enable the SNMP trap on a component, you must update the `Notifications` table in the MyID database to enable notifications and set the destination of the IP address of the external monitoring system.

The following notifications are used:

- Certificate Server Status
- KeyServer Status

For example:

```sql
UPDATE [Notifications] SET [Enabled] = 1, [Destination] = '192.168.1.1'
WHERE [Name] = 'Certificate Server Status' AND [Type] = 'SNMP';
UPDATE [Notifications] SET [Enabled] = 1, [Destination] = '192.168.1.1'
WHERE [Name] = 'KeyServer Status' AND [Type] = 'SNMP';
```

Replace `192.168.1.1` with the IP address of your external monitoring system.

3.3.4   Email notifications

In addition to the notifications with a `Type` of `SNMP`, for certificate server status notifications MyID provides a backup notification with a `Type` of `EMAIL` – this notification contains the same information. If you have configured your system to send email (see section *5*, *Setting up email*) MyID will send email notifications at the same time as it sends SNMP trap notifications. To enable these notifications, use the following SQL statements on the MyID database:

```
UPDATE [Notifications] SET [Enabled] = 1
WHERE [Name] = 'Certificate Server Status' AND [Type] = 'EMAIL';
```

These email notifications use the following email template, which you can edit using the **Email Templates** workflow:

- Certificate Server Status (ID 156)

This email template contain substitution codes for the appropriate error messages.

3.3.5   SNMP Agent notifications

MyID provides an SNMP Agent to allow you to use an external system to monitor the status of MyID's systems. By default, this agent is disabled.

To set up the SNMP Agent, you must edit the following file on the MyID application server:

`C:\Program Files\Intercede\MyID\Components\SnmpAgent\myid.config`

You can set the following options:

Name	Description	Default value
IP	The IP address on which the agent should listen.	127.0.0.1
Port	The port on which the agent should listen.	161
SNMPVersions	A comma separated list of SNMP versions that are supported. Possible values: 1 – Version 1. 2 – Version 2c. 1,2 – Versions 1 and 2c. **Note:** Only SNMP versions 1 and 2c are currently supported.	2
Community	The SNMP community value. This must match the value configured on the monitoring software.	intercede
PrivacyUsername	Reserved for future use.	
PrivacyPassword	Reserved for future use.	
PrivacyAlgorithm	Reserved for future use.	
AuthAlgorithm	Reserved for future use.	
AuthPassword	Reserved for future use.	

Once you have saved the configuration file, you must enable the monitoring for each component in the registry on the MyID application server.

In the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Intercede\Edefice\SNMP
```

there is a single DWORD value for each component. Set the value to `1` to enable monitoring, and `0` to disable monitoring for this component.

3.3.6 Notification reminders

The initial notification is sent when the monitoring service registers the problem. If the problem is not resolved, the notification is sent again after the following intervals:

- 10 minutes
- 30 minutes
- 1 hour
- 4 hours
- 12 hours
- 1 day

After the final notification is sent after 1 day, no further notifications are sent.

4 Database configuration

MyID uses a SQL Server database to store its content. You can also create additional databases for the following purposes:

- Separate audit database
- Archive audit database
- Archive system events database
- Separate image database

MyID uses an additional SQL Server database to store authentication information, including details of audited authentication attempts. You can use this database for reporting; see the *Reporting on the authentication database* section in the ***MyID Authentication Guide*** for details.

You must also consider setting up database maintenance plans and stored procedures for scheduled certificate revocation operations.

4.1 Creating an archive database

You can use the MyID installation program to create an archive database; an archive database is an empty copy of the MyID database that you can use for several purposes, not just as an archive, including:

- A database for archiving the audit trail, server events, or jobs.
- A separate audit database.
- A separate binary objects database.

To create an archive database:

1. Back up the UDL files used for MyID.

   These files are stored in the Windows `System32` folder. Back up the following files:

   - `<databasename>.udl`

     where `<databasename>` is the name you provided in the installation for the main MyID database ; for example, `MyID.udl`.

   - `<databasename>archive.udl`

     For example, `MyIDarchive.udl`.

   - `<databasename>audit.udl`

     For example, `MyIDaudit.udl`.

   - `<databasename>auth.udl`

     For example, `MyIDauth.udl`.

   - `<databasename>binary.udl`

     For example, `MyIDbinary.udl`.

   - `import.udl`

   - `importarchive.udl`

   - `importaudit.udl`

2. Run the MyID installation program.

The MyID installation program is in the Installer folder in the installation media; for example:

```
Installer\MyIDServer-12.10.0.exe
```

Note: If you need more than one additional database (for example, an audit database, an archive database, and a binary database) you must run the installer from a PC that does not have MyID installed. You can create only one archive database from each installation.

In this case, you do not want MyID installed on the PC, as you do not want to change any of the installed components – all you want to do is use the **Archive Database Server** option to create an additional database.
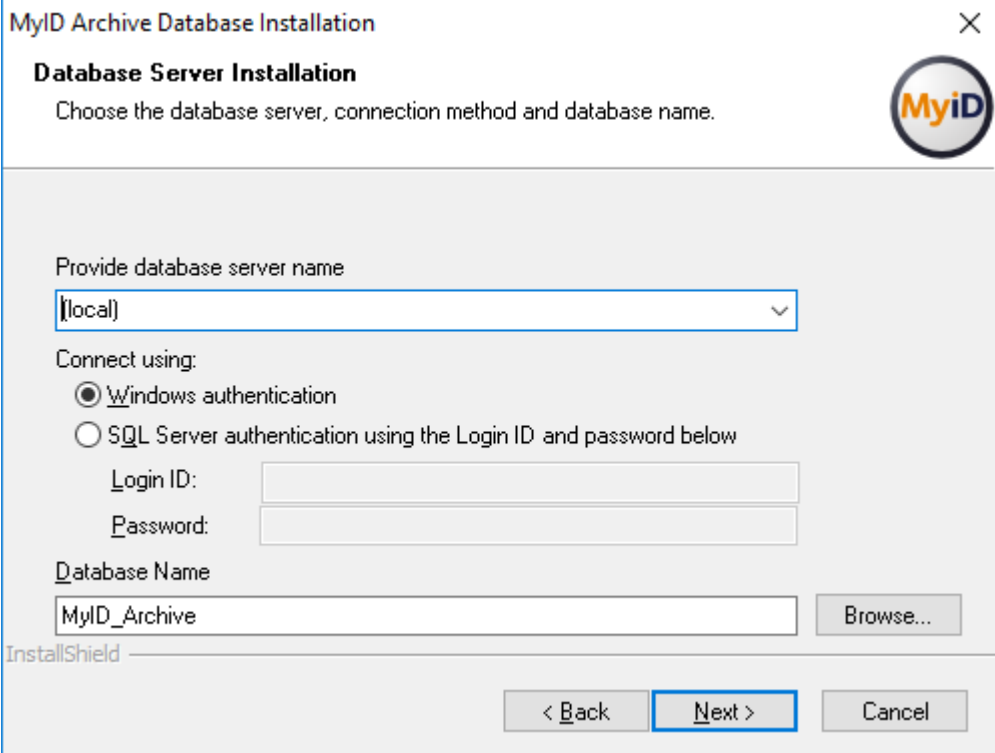
3. At the Server Roles and Features screen, select the **Archive Database Server** option:
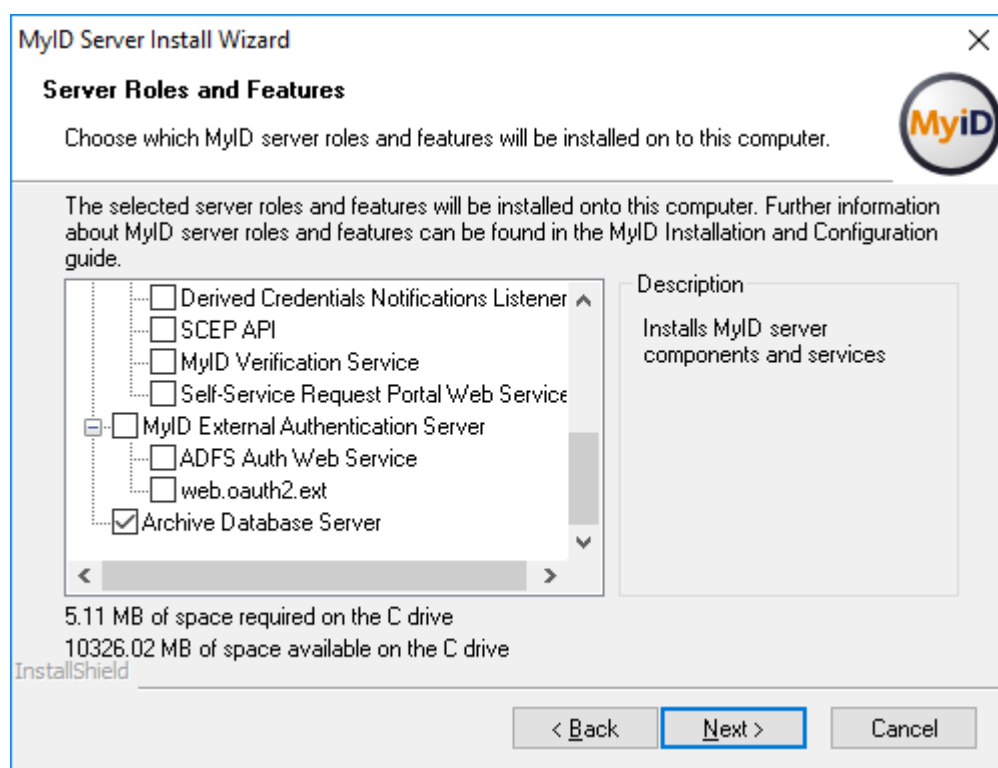
Note: If you are modifying an existing installation, do not change any of the other options. If you are running the installation from a PC that does not have MyID already installed on it, do not select any of the other options.

4. Click **Next**.

5. Type the **User Name** and **Password** for the MyID COM user, then click **Next**.

The MyID Archive Database Installation screen appears:

6. Select the **Database Server** from the drop-down list.

   You may want to store your archive database on a different server from the main MyID database.

7. Type the **Database Name** for the new database.

8. Click **Next**.

   Important: When the installation has completed, if you have any updates installed on your system (that is, patches, hotfixes, and configuration updates), you must apply them to the new database to ensure that the structure and operation data of the additional database matches the main database.

4.1.1 Configuring the data link files

If you run the installation program from the MyID application server, it updates the following Universal Data Link (UDL) files to point to the new database:

- `<databasename>archive.udl`

  where `<databasename>` is the name of the main MyID database ; for example, `MyIDarchive.udl`

- `importarchive.udl`

If you are creating the database for an archive database to store your archived audit trail, system events, or jobs, and you run the installation program from the MyID application server, you do not need to carry out any further manual configuration.

If you run the installation program from the MyID application server, and you are creating the database for another purpose (that is, for an audit database, or a binary objects database) you must reset these UDL files to point to the database you previously used; you can restore the files from the backup you took before you ran the installation program to create the archive database.

If you run the installation program from a PC other than the MyID application server, or you are creating the database as an audit or binary objects database, you must update the appropriate UDL files on the application server to point to the new database:

- Archive database – update the following UDL files:
  - `<databasename>archive.udl`

    where `<databasename>` is the name of the main MyID database ; for example, `MyIDarchive.udl`
  - `importarchive.udl`

- Audit database – update the following UDL files:
  - `<databasename>audit.udl`

    where `<databasename>` is the name of the main MyID database ; for example, `MyIDaudit.udl`
  - `importaudit.udl`

- Binary objects database – update the following UDL file:
  - `<databasename>binary.udl`

To update a UDL file:

1. On the MyID application server, open a Windows command prompt as an Administrator.
2. Navigate to the Windows `System32` folder.
3. Type the name of the appropriate `.udl` file, then press Enter.

   This opens the Data Link Properties dialog, which allows you to change the data link file.
4. Set the properties to point to the server and database you created.

4.2   Using a separate audit database

If you want to store the audit information in a separate database from the main MyID database, you can set up MyID to use a dedicated audit database.

To create a separate audit database, run the MyID installation program to create an archive database. See section *4.1*, *Creating an archive database* for details. Once you have created the new empty database, you must configure the following data link files to point to the new database:

- `<databasename>audit.udl`

  where `<databasename>` is the name of the main MyID database ; for example, `MyIDaudit.udl`
- `importaudit.udl`

See section *4.1.1*, *Configuring the data link files* for details.

4.3 Archiving the audit trail

Every operation within MyID generates audit information. Over time, this information can build up and ultimately reduce performance due to the size of the data. As a solution to this, you can create an *archive* database.

- Old audit data (which is likely to be rarely viewed) is transferred into this database using a SQL scheduled task.

  Note: MyID does not automatically archive records; you must set up the scheduled SQL task. See section *4.3.2*, *Create a SQL timed task* for details.

- The quantity of the *live* (recent) audit data, which is the data that is viewed most often, is kept small. This improves search performance.

Warning: The instructions in this section allow you to archive the audit trail. You must check Microsoft documentation for full operating instructions for Microsoft SQL Server.

You can still view archived audit information within MyID. The default MyID installation stores both current and archived audit information (separately) within the main MyID database. You can also store current and archived audit information in separate databases, which could be on separate servers.

4.3.1 Create a separate database for archiving audit records

The difference between an archive database and a separate audit database is that the audit database is used for *current audit records* and uses the `MyIDaudit.udl` file to point to the database, and the archive database is used for *old* audit records, and uses the `MyIDarchive.udl` file to point to the database.

To create a separate archive audit database, run the MyID installation program to create an archive database. See section *4.1*, *Creating an archive database* for details. Once you have created the new empty database, you must configure the following data link files to point to the new database:

- `<databasename>archive.udl`

  where `<databasename>` is the name of the main MyID database ; for example, `MyIDarchive.udl`

- `importarchive.udl`

See section *4.1.1*, *Configuring the data link files* for details.

4.3.2 Create a SQL timed task

This procedure is performed on the SQL server that stores the live audit information.

1. Select the **SQL Server Agent** in the Microsoft SQL Server Management Studio.

   Note: You will not be able to access the folders described in the following steps unless the service is running. You may have to start it by right-clicking it and selecting **Start** from the menu.

2. Right-click the **SQL Server Agent** and select **New**, then **Job** from the menus displayed.
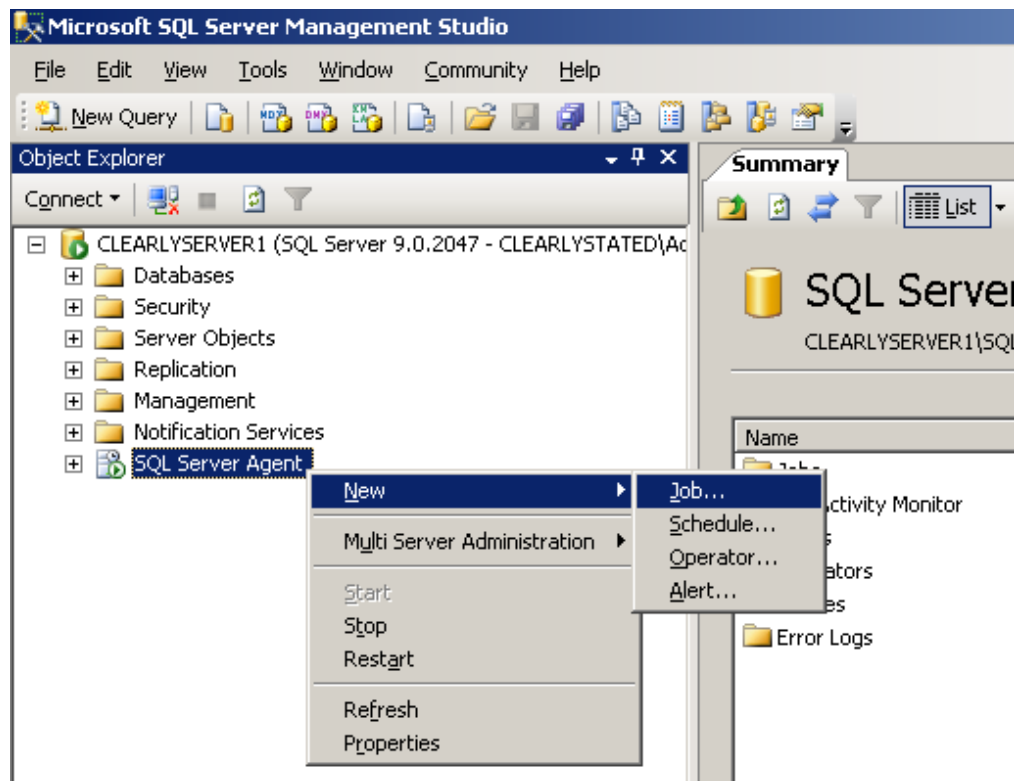
3. The **New Job** box is displayed, with the **General** page highlighted.

   a. Give the job an appropriate **Name** to help you identify it later.

   b. Set **Owner** to an account with administrative privileges.

   Note: If the archived data is to be stored on a separate server, the account must have sufficient privileges for both the current server *and* the server that will be used to store the archive.

4. On the **Steps** page, click **New** to create a new step for the job.

   The **New Job Step** box is displayed.

   a. Enter a **Step name**.

   b. Select the **Database** that contains the data to be archived; this is your main MyID database.

5. In the **Command:** area, type:

   ```
   sp_ArchiveAudit '<archivedatabase>', <daysOld>
   ```

   where:

   - `<archivedatabase>` is the name of the database that will store the archived data.

     - If a single database is being used to store both live and archive information, then the value of `<archivedatabase>` will be the same as the name selected from the list in **Database**. The archived data will be moved into a separate table.

     - If the archive database name begins with numbers, you must enclose the database name in square brackets. For example:

```
sp_ArchiveAudit '[20100101_CMSArchive]', 90
```

- If the archive database exists on a different server, you must configure this as a named "linked server" within SQL Enterprise manager.

  The `<archivedatabase>` would then be specified as:

  ```
  <LinkedServerName>.<ArchiveDatabaseName>
  ```

  where:

  - `<LinkedServerName>` is the name of the linked server.

  - `<ArchiveDatabaseName>` is the name of the archive database on that server.

- `<daysOld>` is the age of data, in days, that will be archived.

  For example:

  ```
  sp_ArchiveAudit 'ArchiveDB', 90
  ```

When this task runs, all audit data that is more than 90 days old is moved from the database chosen in **Database** to the `ArchiveDB` database.
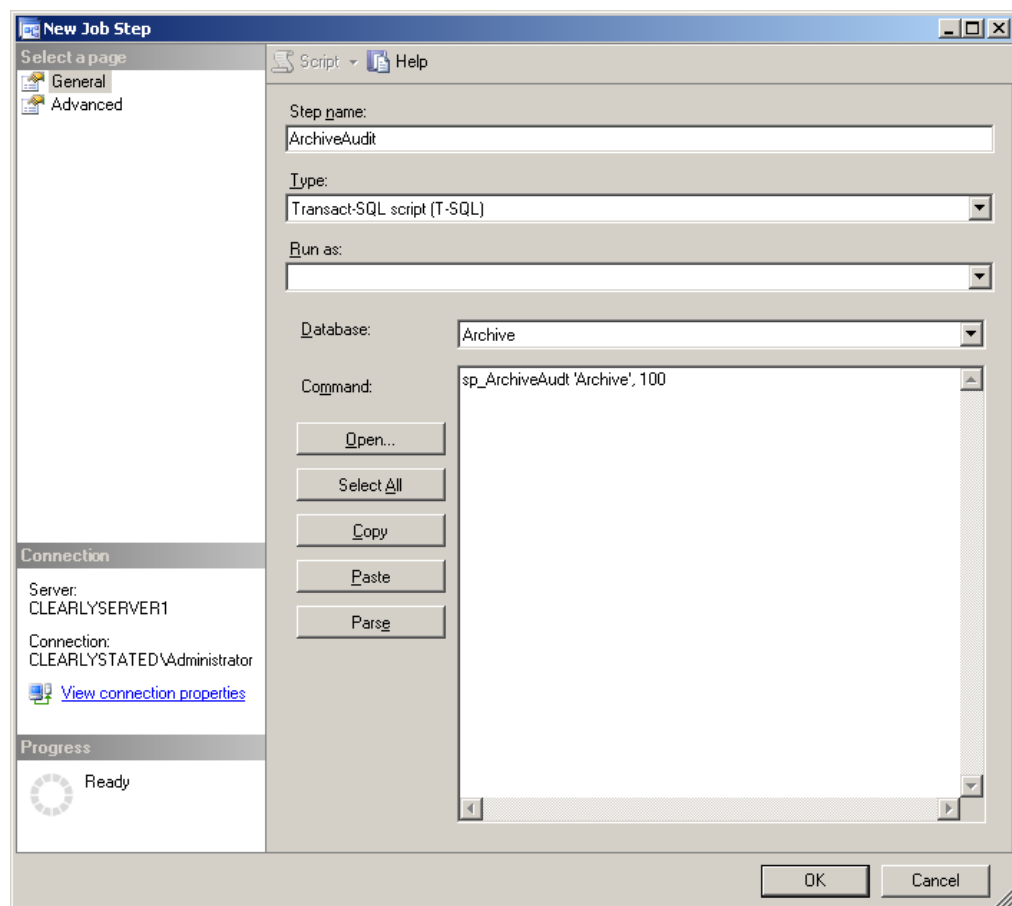
6. On the **Advanced** page, specify a log file.

The audit archiving procedure produces a log file that reports statistics concerning the archiving procedure, including the number of records archived and error information. Intercede recommends that you record this information in a log file as a record of the archiving procedure.

  a. Enter a name for the log file in the **Output file** box.

   Make sure that the path specified is a valid directory on the SQL server.

  b. If you want to keep earlier information, select **Append output to existing file**.

   If you do not select **Append output to existing file**, the log file will be replaced every time the archive procedure runs and you will lose earlier information unless you have taken other steps to retain it.

  c. Select the **Include step output in history** option.

  d. Click **OK**.

7. Create a schedule.

  a. Click **Schedules**.

  b. Click **New**.

   Although it is possible for users to access MyID while archiving is taking place, it is better to select an off-peak time (for example, overnight) so database performance is not affected.

c. Give the schedule an appropriate **Name**.

d. Select **Recurring** in **Schedule type**.

e. Make sure the **Enabled** box is selected.

f. Set a schedule to meet your requirements. The example shows a daily schedule, at 3:00 a.m.

g. Click **OK** on the **New Job Schedule** dialog box to accept the schedule.

h. A summary of the schedule is displayed. Click **OK**.

8. Click **Notifications**.

a. Select **Write to the Windows Application event log** and **When the job completes** from the associated drop-down list.

You may optionally specify that operators be emailed or paged according to your own administrative policies. This would allow you to further track the status of the archiving.

b. Click **OK**.

The timed archiving task is now configured. Check the logs to make sure that the audit archive procedure is running successfully.

4.4 Archiving the System Events

To create an archive database to store your system events, run the MyID installation program to create an archive database. See section *4.1*, *Creating an archive database* for details. Once you have created the new empty database, you must configure the following data link files to point to the new database:

* `<databasename>archive.udl`

  where `<databasename>` is the name of the main MyID database ; for example, `MyIDarchive.udl`

* `importarchive.udl`

See section *4.1.1*, *Configuring the data link files* for details.

You can set up MyID to archive the contents of the system events table in the MyID database periodically in a similar way to archiving the `Audit` table; see section *4.3*, *Archiving the audit trail*.

You can use the `LogEventsArchive` table, and a stored procedure, `sp_ArchiveLogEvents`.

Set up a SQL Timed Task on your MyID database to run the `sp_ArchiveLogEvents` procedure periodically. The syntax is as follows:

```
sp_ArchiveLogEvents '<archivedatabase>', <daysOld>
```

where:

* `<archivedatabase>` is the name of the database that will store the archived data.

  * If a single database is being used to store both live and archive information, then the value of `<archivedatabase>` will be the same as main MyID database. The data will be moved into a separate table.

  * If the archive database name begins with numbers, you must enclose the database name in square brackets. For example:

    ```
    sp_ArchiveLogEvents '[20100101_CMSArchive]', 90
    ```

  * If the archive database exists on a different server, you must configure this as a named "linked server" within SQL Enterprise manager.

    The `<archivedatabase>` would then be specified as:

    `<LinkedServerName>.<ArchiveDatabaseName>`

    where:

    * `<LinkedServerName>` is the name of the linked server.

    * `<ArchiveDatabaseName>` is the name of the archive database on that server.

* `<daysOld>` is the age of data, in days, that will be archived.

  For example:

  ```
  sp_ArchiveLogEvents 'ArchiveDB', 90
  ```

4.5 Archiving jobs

The `Jobs` table in the MyID database is used for managing active issuance jobs as well as retaining a history of completed jobs. Over time, the information on completed jobs can build up and ultimately reduce performance due to the size of the data. As a solution to this, you can archive the completed, failed, and canceled jobs over a certain age.

- Old job data (which is likely to be rarely viewed) is transferred into a separate table using a database stored procedure.

  Note: MyID does not automatically archive records; you must either run the procedure manually or set up a scheduled SQL task. See section *4.5.2*, *Running the stored procedure* for details.

- The quantity of the live (recent) job data, which is the data that is viewed most often, is kept small. This improves search performance.

Warning: The instructions in this document allow you to archive the job information. You must check Microsoft documentation for full operating instructions for Microsoft SQL Server.

You can store the archived job information in the main MyID database, or alternatively you can configure a separate database to store this information.

4.5.1 Setting up a separate database for the jobs archive

To create a job archive database, run the MyID installation program to create an archive database. See section *4.1*, *Creating an archive database* for details. Once you have created the new empty database, you must configure the following data link files to point to the new database:

- `<databasename>archive.udl`

  where `<databasename>` is the name of the main MyID database ; for example, `MyIDarchive.udl`

- `importarchive.udl`

See section *4.1.1*, *Configuring the data link files* for details.

Once you have created the archive database, you must configure it for archiving jobs.

1. In the main MyID database, edit the `ArchiveDatabaseLocation` function:

   a. Locate the following:

   ```
   ALTER FUNCTION [dbo].[ArchiveDatabaseLocation] ( )
   RETURNS sysname
   AS
   BEGIN
       DECLARE @Location sysname
       select @Location = db_name()
       RETURN @Location
   END
   ```

   b. Change the function to return the name of your archive database instead of the database; for example:

   ```
   ALTER FUNCTION [dbo].[ArchiveDatabaseLocation] ( )
   RETURNS sysname
   AS
   BEGIN
       RETURN 'MyIDArchive'
   END
   ```

   c. Run the query to update the function.

   d. Carry out one of the following:

   - Re-run the installation procedure to install the main MyID database again.

     Note: The installation program does not allow you to re-run the database installation without uninstalling the database component first. Instead, you can run the installer from a PC that does not have MyID installed, and select only the database option; this re-runs the scripts against the existing database.

   - If you are using Project Designer to customize your system, re-run the Project Designer scripts.

   This updates the following views in the MyID database:

- `mis_PIVArchivedRequests`
- `mis_PIVAllRequests`

These views are used for the Archived Requests and All Requests reports in the MyID Operator Client, to allow the reports to include information from the archive database.

Note: You do not have to update the function again if you upgrade MyID. You need to update the function only if you change the name of the archive database at a later date.

Note: If you add a separate jobs archive database after initially storing your archived jobs in the main MyID database, the stored procedure does not copy the archived jobs from the main database to the archive database; you must migrate this data manually.

4.5.2 Running the stored procedure

This procedure is performed on the SQL server that stores the live job information. You can either run the procedure manually or set up a timed task; see your Microsoft documentation for details of creating an SQL timed task.

The syntax of the stored procedure is:

```
sp_ArchiveJob '<database>', <daysOld>
```

where:

- `<database>` is the name of the MyID archive database.

  This can be either the main MyID database or a separate MyID archive database.

  If the database name begins with numbers, you must enclose the database name in square brackets. For example:

  ```
  sp_ArchiveJob '[2013_mydatabase]', 90
  ```

- `<daysOld>` is the age of data, in days, that will be archived.

  For example:

  ```
  sp_ArchiveJob 'mydatabase', 90
  ```

When this procedure runs, all completed, failed or canceled job data that is more than 90 days old is moved from the jobs table to the job archive table.

4.5.3 Testing the job archive process

You can use the following stored procedure to test the job archive process:

```
sp_archiveJobCopy
```

This stored procedure operates in the same way as `sp_ArchiveJob`, but does not remove job data from the `Jobs` table after copying it to the archive table.

4.5.4 Database views

You can use the following views to assist in reporting data from the primary and archive job tables:

- `vJobsArchive` – a replica of `vJobs` using the archived jobs data only.

- `vJobsAndArchive` – reports `Jobs` table data from live and archive tables.

- `vJobsExAndArchive` – reports `JobsEx` table data from live and archive tables.

Note: These views return data from the current database only. They do not access information in a separate jobs archive database.

You can also use the following views, which provide a more limited set of fields, but are designed to include data from both the main MyID database and the archive database, if configured:

- `mis_PIVArchivedRequests` – contains archived jobs data only.

- `mis_PIVAllRequests` – contains both live and archived jobs data.

4.5.5 Viewing archived jobs

The Archived Requests and All Requests reports in the MyID Operator Client allow you to view request jobs that have been archived.

If you are using a separate archive database, and you do not see archived information in these reports, make sure you have updated the `ArchiveDatabaseLocation` function and the `mis_PIVArchivedRequests` and `mis_PIVAllRequests` views; see section *4.5.1*, *Setting up a separate database for the jobs archive* for details.

See the *Archived Requests report* and *All Requests report* sections in the ***MyID Operator Client*** guide for details of running the reports.

4.6 Creating a separate database to store images

You can set up MyID to store the binary objects such as images in a separate database from the main MyID database. This works in a similar way to setting up a database for archived audit data.

To create a separate binary objects database, run the MyID installation program to create an archive database. See section *4.1*, *Creating an archive database* for details. Once you have created the new empty database, you must configure the following data link file to point to the new database:

- `<databasename>binary.udl`

    For example, `MyIDbinary.udl`.

See section *4.1.1*, *Configuring the data link files* for details.

Once you have created the database, you must set up permissions on the database for the MyID named COM+ user:

1. In Microsoft SQL Server Management Studio, expand the **Security** folder and select **Logins**.

2. Right-click the service account user and select **Properties**.

3. In the **Select a page** section, click **User Mapping**.

4. Select the database you created to hold the binary objects.

5. Grant the account the following roles for the database:

    - `public`

    - `db_owner`

    - `db_datareader`

    - `db_datawriter`

6. Click **OK**.

4.7 Creating database maintenance plans

A maintenance plan should form part of your database authority recovery procedures.

Enterprise Manager provides a method to create maintenance procedures to manage the size of the transaction log and database files. It is important that these procedures are adopted following installation.

The maintenance plan is activated as a wizard within Enterprise Manager. You can either:

- Select **Taskpad** from the **View** menu and choose **Create a Maintenance Plan** from the list of wizards provided.

- Right-click the database, select **All Tasks** and then **Maintenance Plan** from the menu displayed.

For a full explanation of maintenance plan procedures, consult the appropriate Microsoft SQL documentation available as part of the SQL installation or online from ***msdn.microsoft.com***.

4.8 Scheduled certificate revocation operations

MyID provides the ability to execute scheduled certificate request and revocation operations. This is typically used to perform regular maintenance tasks, such as automatically revoking certificates that have been suspended for a preconfigured length of time.

The detection and flagging of certificates to be revoked is typically performed by a stored procedure (for example, `sp_CertStatusRevokeProcess`). The submission of these requests to the Certificate Authority relies on processes carried out automatically by the Certificate Services, which are set up during installation.

Note: To configure MyID to revoke suspended certificates after a given time period, you have to edit the **Suspend to revoke period** option in the **Certificates** tab of the **Operation Settings** workflow. Update the value to the number of days a suspended certificate exists before revocation. By default, this option has a value of zero, which will be ignored by the automatic processor.

5 Setting up email

MyID allows you to configure SMTP servers to send email messages.

You can create more than one SMTP server – MyID will send email notifications through each configured external system.

Note: Previous versions of MyID used Database Mail. For more information on upgrading existing systems, see the *Upgrading email support* section in the *Installation and Configuration Guide*.

To set up an SMTP server:

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

   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. On the **Notifications** tab, set the **Send Email Notifications** option to Yes.

   Note: You may have to restart the MyID Notifications Service to pick up this change.

3. Click **Save changes**.

4. From the **Configuration** category, select **External Systems**.

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

5. Click **New**.

6. From the **Listener Type** drop-down list, select **SMTPServer**.

7. Set the following options:

   - **Name** – Type a name for the external system.

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

   - **Enabled** – Make sure this option is set to Yes ✓.

- **SMTP Server** – Type the IP address or name of the SMTP server. For example:

  `smtp.example.com`

- **SMTP Port** – Type the port number of the SMTP server. For example, `25`.

- **SMTP Authentication** – Select one of the following options:

- **Anonymous** – The SMTP server does not require any authentication.

- **Use application account** – Authenticate to the SMTP server using the MyID named COM user.

- **User/password authentication** – Type the SMTP Username and enter and confirm the SMTP Password to be used to authenticate to the SMTP server.

- **Use TLS for SMTP** – Select this option if you want MyID to connect to the SMTP server using a secure (TLS/SSL) connection.

- **MyID Mail From address** – Type the email address that will appear in the From field on email messages sent by MyID.

- **MyID Mail Reply-To address** – Type the email address that will be used when a user replies to an email message sent by MyID.

- **Sign outgoing emails** – Select this option to sign the content of the email messages that MyID sends. See section *5.1*, *Signing email messages* for details.

- **Email address (for test connection)** – Type an email address that will be used when you click the **Test connection** button.

  Note: The email address for the test connection is not stored when you save the external system.

8. Click **Test connection**.

   MyID sends a test email message to the specified email address. Check that the email message has been received.

   Note: You cannot send a test email message if the **Send Email Notifications** option is set to `No`. Also, if you are using signing, and have multiple application servers, this test will confirm that the signing certificate is set up only on the application server to which you are currently connected.

9. Click **Save**.

5.1   Signing email messages

MyID can sign the content of the email messages it sends. You must make sure that you have set up the following:

- Set up the certificate template on the certificate authority to include the **Secure Email** attribute in the **Application Policies** extension.

  Note: If you do not set this attribute on the certificate template, the email messages will be sent, but will be unsigned.

- Configure the MyID application server that is processing the email with a valid signing certificate.

To configure the application server's signing certificate:

- Import or create an email signing certificate where the Subject matches the From address of the SMTP configuration.

- Export the email signing certificate to a `.cer` file on the application server.

- Set the following registry value to the full path of the `.cer` file on the application server:

```
HKEY_LOCAL_
MACHINE\SOFTWARE\Intercede\Edefice\Server\Mail\SigningCertificate
```

- Set the **Sign outgoing emails** option for the SMTP server in the **External Systems** workflow.

- Set the **Signed** option for the appropriate email template in the **Email Templates** workflow.

6 Business continuity planning

MyID is designed to be a critical part of enterprise security architecture and needs to be fully integrated into a disaster recovery plan. The critical part of any disaster recovery system is advanced preparation.

In practice each system deployment is different, with its own characteristics. While this document can provide a high-level outline plan, we recommended that you contact Professional Services at Intercede to design a full disaster recovery capability.

6.1 Phase 0: pre-disaster

Ensure you have backed up the live data.

- Backup the MyID databases on the existing live server.

- Backup the MyID data in the `upimages` folder on the live web server, if necessary.

- Backup the MyID windows registry information on the live application server.

- Ensure any security credentials are backed up. For example: authentication or signing credentials to access a PKI system, or HSM data.

- Store copies of the MyID application software, plus any local customization and patches, in a secure location, off-site.

6.2 Recovery

Disasters can happen on many levels, from losing just one disk on one server through to an entire server infrastructure being lost. This section describes rebuilding a full three-server MyID installation. You can use portions of this information if the disaster is confined to a specific server.

Warning: Before attempting any system recovery, we recommend that you contact Intercede's support team to validate your recovery plan.

6.3 High-level recovery plan for re-building a three-server architecture

Note: This section assumes a standard three-server architecture without advanced configuration such as a DMZ.

6.3.1 Phase 1: Prepare new servers

1. Install Windows Server on three new servers: web, application and database.

   - SQL Server should be installed on the database server.

   - All three servers should be members of the same Windows domain.

   - Ensure that the time on the new database server is synchronized with the time on the previous database server.

2. Restore any third party systems that MyID requires access to, for example a directory or certification authority (CA).

3. Restore any credentials needed to access the third party systems: for example PKI keys, HSMs.

4. Install the MyID web, application and database components on the appropriate servers. Follow the installation documentation but incorporate any site-specific deviations that you made for the original installation.

5. Install any local customizations and patches.

6.3.2 Phase 2: Restore backed-up data

1. Restore data from the `upimages` folder on the live web server to the web server.

   If you are using an external server as an image store, you may not need to carry out this step.

2. Restore the MyID registry files from the live application server to the new application server. This will overwrite the existing registry settings, with the appropriate key information relating to the backed up database.

3. On the database server, replace the newly installed MyID databases with the backed up database from the live system.

4. Replay any transaction log data from the live system to the new database to ensure data is restored up until the point of failure.

5. Check the **Configuration** tab on the **System Status** report and update any configuration options that refer to specific server names or IP addresses.

6. Reboot the three new servers to ensure they are using the new registry and databases.

6.3.3 Phase 3: Test new system

1. Test basic MyID operations. For example, can Operators logon with their smart cards.

2. Test end-to-end card production.

3. Assuming everything is functioning correctly, backup the new MyID database.

6.4 Two-server and one-server architectures

The procedure for two-server and one-server architectures is essentially the same as for a three-server architecture. Apply the appropriate steps to the relevant co-located server.

6.5 System integration

MyID is designed to interact with third party systems, such as directories and certification authorities. Depending on the scope of the disaster, these systems may need to be restored as well.

This introduces a complexity in the recovery timing that needs to be carefully planned to ensure the security integrity of the overall system. Care needs to be taken to ensure that the recovered system has end-to-end data integrity.

For example consider a three-component system, containing a directory, a CA and MyID. To issue a PKI credential, the user must be known to all three components.

In a disaster where data recovery is required, it is important to ensure that MyID, the directory and the CA are all restored to exactly the same point in time. If not:

- The directory may contain users not in MyID, if the database was recovered from a later period than MyID.

- The CA may have records of certificates issued that are not known to MyID and so cannot be revoked by MyID. This would happen if the CA was recovered from a later period than MyID.

- MyID may have knowledge of credentials issued that are not known to the CA, if the CA was recovered to a period of time before MyID. This is a critical security issue as in principle a live certificate has been issued by MyID that cannot be revoked, as the CA has no knowledge of it.

7 Failover strategy

MyID has been designed to support failover and redundant component architectures to ensure the availability of the MyID solution. This section describes the architecture options available.

7.1 Typical MyID architectures

MyID consists of three major system components: a web server, an application server and a database server.

These can be hosted on:

- A single machine

- Two machines (separating the database from the other two components)

- Three separate machines

Typically a two-machine architecture is used, with the web server and application server co-hosted on a single machine:
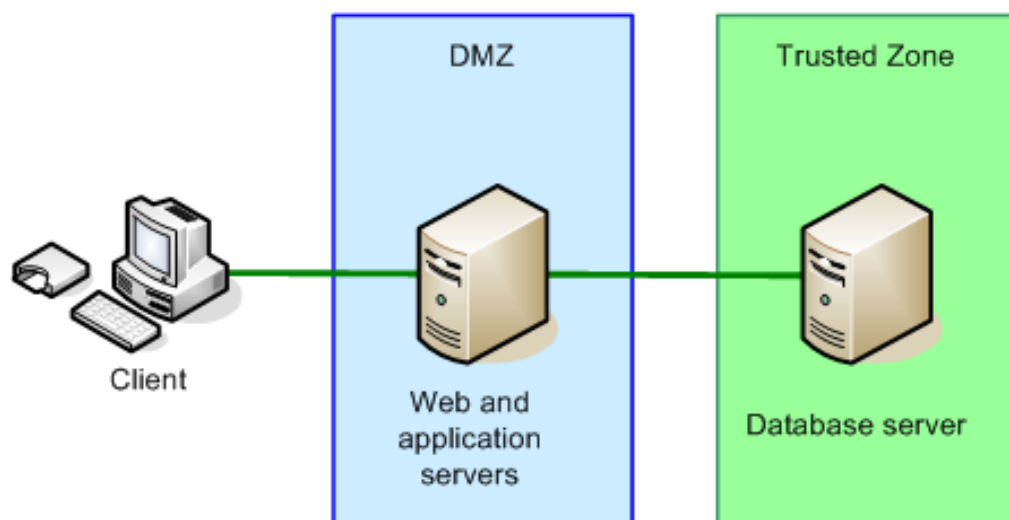
For high security environments, a three-machine architecture may be used:
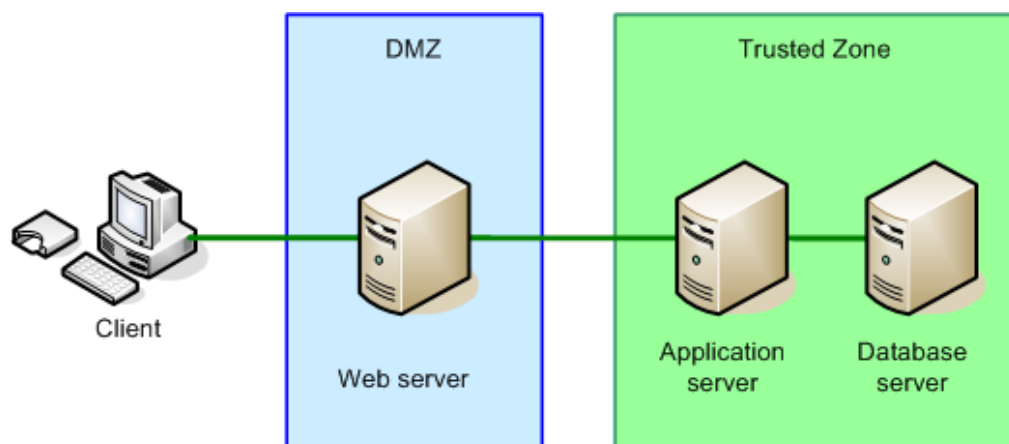
7.2 Co-hosted web and application servers

7.2.1 Duplicate infrastructures

One option is to have a duplicate MyID infrastructure, with a failover TCP/IP router providing access.

- If the web server fails, the router will automatically switch traffic to the backup website.

- If the database server fails, a manual switch of the failover router would be required. There are many commercially available products for monitoring systems that can do this automatically.

- The backup database contains a duplicate of the main database, provided using the Microsoft SQL Server Transaction Log Shipping system, for example.

    Other systems that can provide database duplication are:

    - Database failover clustering

    - Database backup / restore procedures

    - Disk / file system mirroring

    - Third party utilities

Note: You cannot have two live databases running at the same time.

During a failover, any current operations will fail and the user will need to re-authenticate before continuing. Other than that the operation should be seamless.

An example of this configuration is shown, with the web server and application server co-hosted on a single machine:
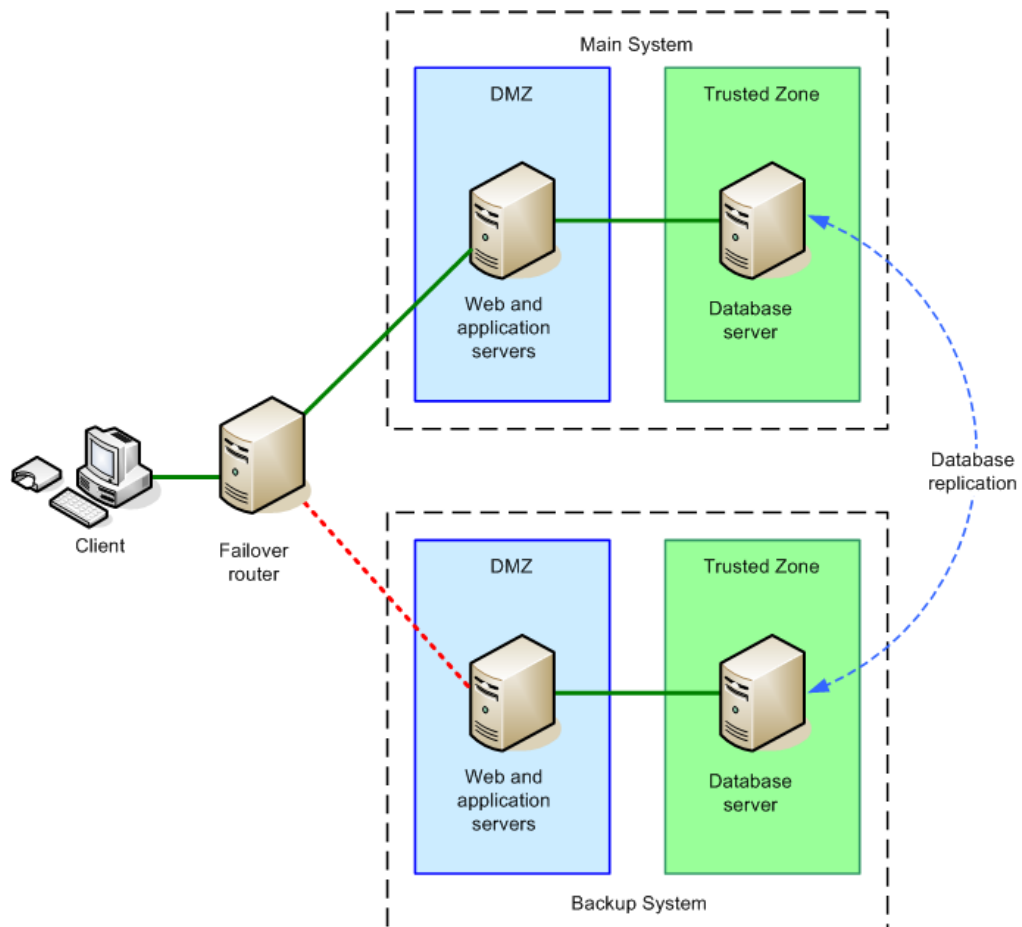
This solution provides resilience against both component and network failure. It is common to locate the main system and backup system on different networks, hosted on different sites, to protect against total infrastructure failure at on site, such as a major power failure.

7.3   Split web and application servers

For additional security, the application and web servers may be hosted on separate machines, on different network segments:
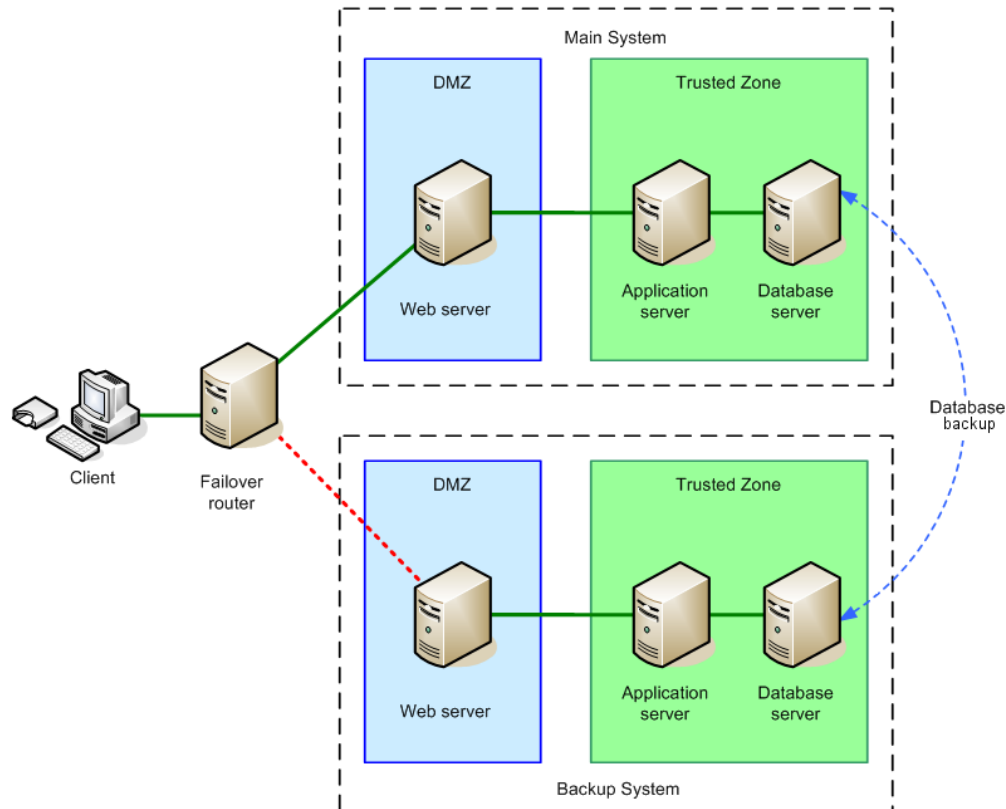
If the database server fails, a manual switch of the failover router would be required. As previously stated, there are many commercially available products for monitoring systems that can do this automatically. Contact customer support for more information.

Note: You cannot mix servers from the main system with the backup system – if the web server from the main system fails, and the application server from the backup system fails, there is no load balancing configuration that will allow the backup web server to work with the main application server.

7.4   Additional considerations

7.4.1   User images

User images are stored on the MyID database server, and are handled by database replication.

Images uploaded from the Card Layout Editor are stored on the web server.

On older systems, user images may also be stored on the web server; in environments where the images are uploaded by the user and duplicate web servers are used, extra configuration will be required to ensure the virtual file store used by the images is available to both servers. A typical option to use is Microsoft's file store synchronization.

Note: You may store your images on a separate server, in which case you must ensure that the image store is available on a duplicate server.

7.4.2 Clustering

The architectures presented in this document so far can be used to build a resilient infrastructure when a failover capability is required to ensure system availability.

In some environments, once redundancy has been built into the system, the backup systems are used to provide load-sharing. This can be achieved using clustering technology.

A cluster architecture will also provide even greater system availability than the failover solutions described above.

This example shows how MyID can be used with a database cluster:
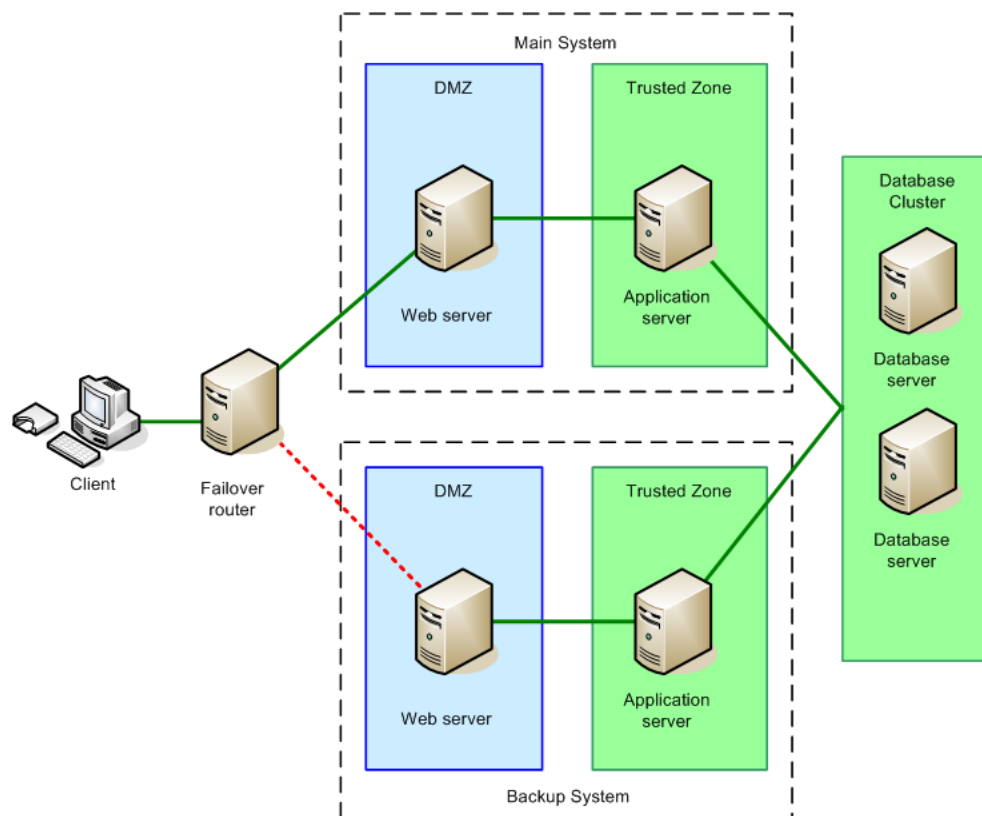
As well as database clusters, web server clusters or farms can be used, provided that support for session affinity is enabled, as MyID uses ASP Session State. You must make sure that your servers support session affinity both for the MyID web servers and the MyID web services servers.

You can also consider using load-balancing across a farm of application servers.

7.4.3 Hardware

All of the servers in the system should have redundant components. Specifically we recommend dual power supplies, a RAID disk array and dual network cards.

Clustering Microsoft Windows servers usually requires dedicated cluster hardware with high-speed connections between the servers. While this is more expensive than 'standard' servers, it provides the highest levels of redundancy and load-balancing.

7.5 Failover and redundancy considerations

The text referring to support for 'web server clusters' means that you can add additional servers for failover/backup purposes for the web layer, but the important thing to understand is that there is an architectural limitation of the COM+ components that are at the core of MyID that means that there must be a fixed mapping between web servers and the application server that they are paired with.

For example, in this diagram on a linear 3-server model, the COM+ proxies that run on the web server are:

- created from the original COM+ components on the application server,

- then exported to the web server and installed there,

- then used by the MyID website to drive the primary components on the application server.
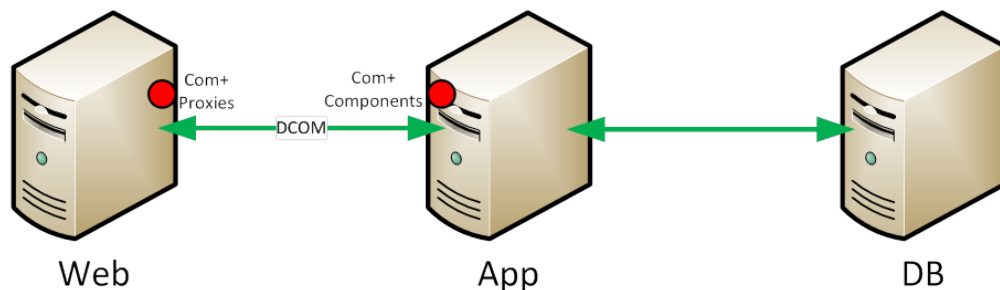
The proxies that run on the web server must be derived from the instance of the components on the individual application server that they are paired with.

This means that you can add additional web servers (for example, a cluster or farm) that share the same proxies and are therefore paired with a specific application server. For example:
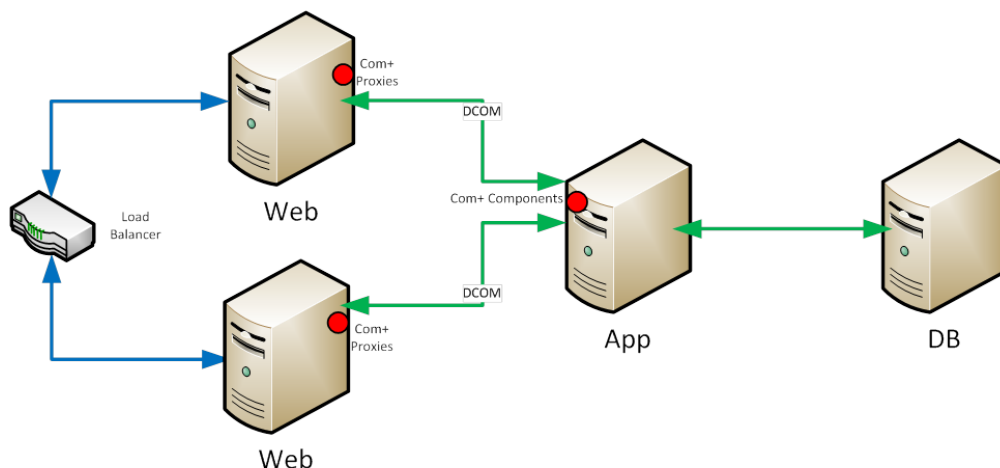
This is OK because each web server (with the COM+ proxies that run it) is still paired with a specific application server. However, this model creates redundancy/failover in the web layer only, meaning that there is still a single point of failure at the application server level.

Therefore, customers seeking a fully redundant system are advised to duplicate the application/web channel, as well as using a SQL Cluster for hosting the MyID database:
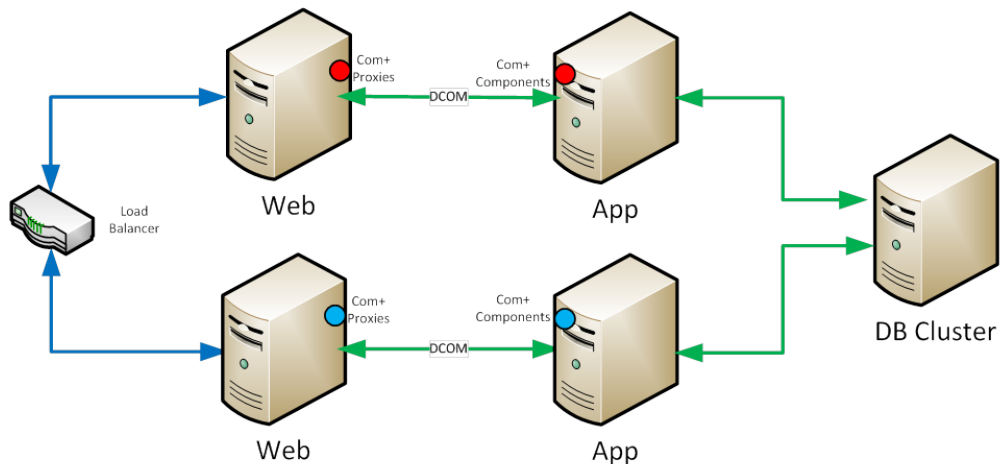
In this model each web server is individually paired with each application server (represented by the different colored circle for the COM+ components).

There cannot be any failover from a web server in one channel to the application server in another without manual reconfiguration. For example:
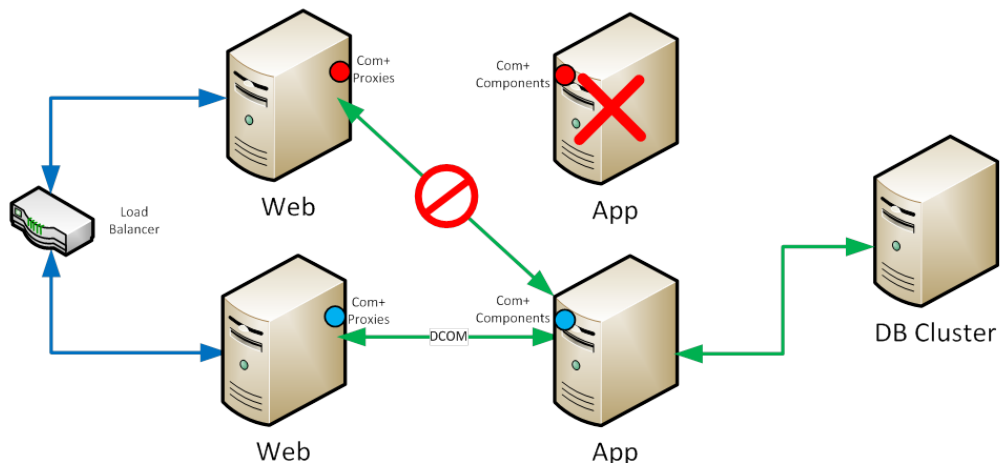
This is because the proxies on the web server in the top channel are not paired with the application server on the second channel. That is, the communications from top channel web server to the second channel application server will not work.

It is for this reason that each web to application channel must be treated as a single unit in failover decisions.

That is, if any part of the top channel fails, the load balancer/failover router can switch all MyID traffic to the second channel:
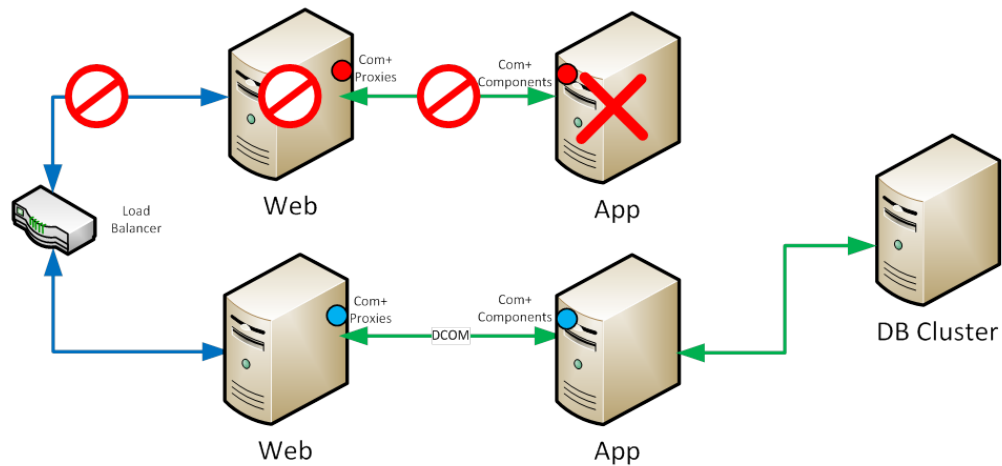
This requires that the load balancer/failover router be capable of monitoring the health of each channel as a whole and not just the web server that sits at the front of it.

This depends on the capabilities of the load balancer/failover router in use. For example, potentially it could base its failover decisions on a combination ping/heartbeat to both the web and application server in each channel.

8　Performance and sizing

MyID is an application that can be deployed in many different configurations dependent upon business requirements.

These deployment models include high data storage solutions (for example, where a 1 million user population is to be managed) and high usage solutions (for example, where 300,000 cardholders are self-collecting certificates concurrently).

Due to the many different ways the system can be deployed and the different systems it can be integrated with to meet a particular project's requirements, it is not possible to have a 'one size fits all' recommended deployment strategy and resultant sizing model; a one million user project issuing cards through a third party bureau may well place less load on the web server than a 10,000 user self-service solution for example.

The purpose of this section is to give information on how MyID should be expected to perform under known circumstances. It is designed to act as a start point to allow partners to calculate the required number and specification of servers for a particular project.

8.1　Performance

Every individual action performed by an operator within MyID creates a 'session' with the server. This session is not a permanent link in that it does not hold onto resources, but a way of associating packets of data received with a particular operation (for example, issue card).

The more operations that need to occur simultaneously, the more memory and processing power is required to perform them.

The following information is based on data retrieved from a large real-world MyID installation and can be used to estimate how many concurrent operations can be managed by a single server.
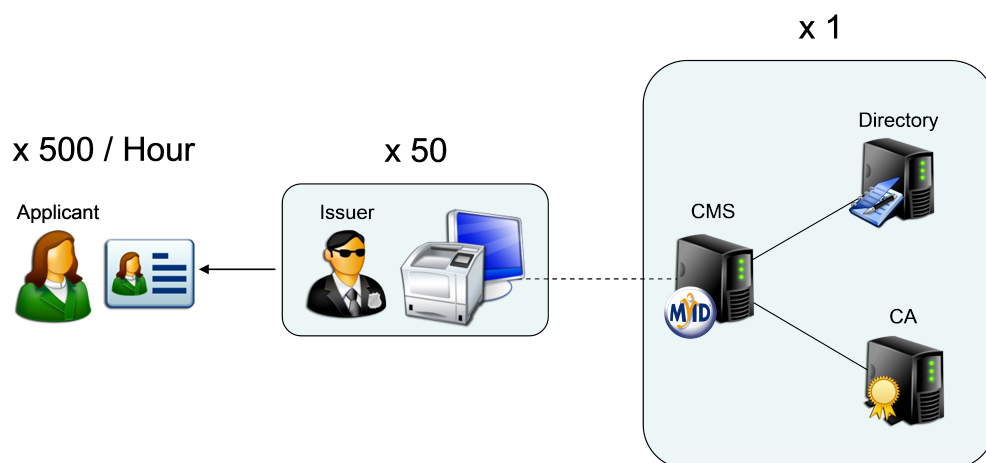
In the example above a simple single server MyID architecture (web server, application server and database all installed on the same machine) was used. MyID was connected to an Active Directory and a CA.

The use case was as follows:

- Log on to MyID with certificate on card.
- Access the issue card workflow.

- Browse to a user in the directory.

- Select a user and choose a credential profile.

- Insert a new card to issue.

- User enters PIN.

- MyID writes two certificates to the card.

- Log off.

It was found a single machine could cope with 50 operators (each connecting from a different client PC) each logging on and issuing cards concurrently. This placed a 75% processor load on the machine and was seen to give negligible performance loss over a single operator usage.

During this time each operator was issuing 10 cards per hour leading to a total of 500 cards per hour (50 operators at 10 cards per hour each) or 1,000 certificates per hour.

8.2 Sizing

Due to the capability of storing information on the form of customized attributes (that is, fields of information held against a group / device or person) and the dynamic content of those attributes it is difficult to predict the exact storage requirements of MyID.

In addition, the ability to define exactly how much information is audited and the fact that different credentials may be used per project (an archived key and certificate will require more storage than a single system wide applet for example) means the best place to calculate the exact storage requirements is from the deployed architecture itself.

In reality, an estimated sizing can be carried out in advance with this estimate being checked during a Proof of Concept / pilot phase of a project (please contact Professional Services for assistance if required).

The following information is based on data retrieved from an actual installation of MyID and can be used to estimate how much data storage is typically required per user record.
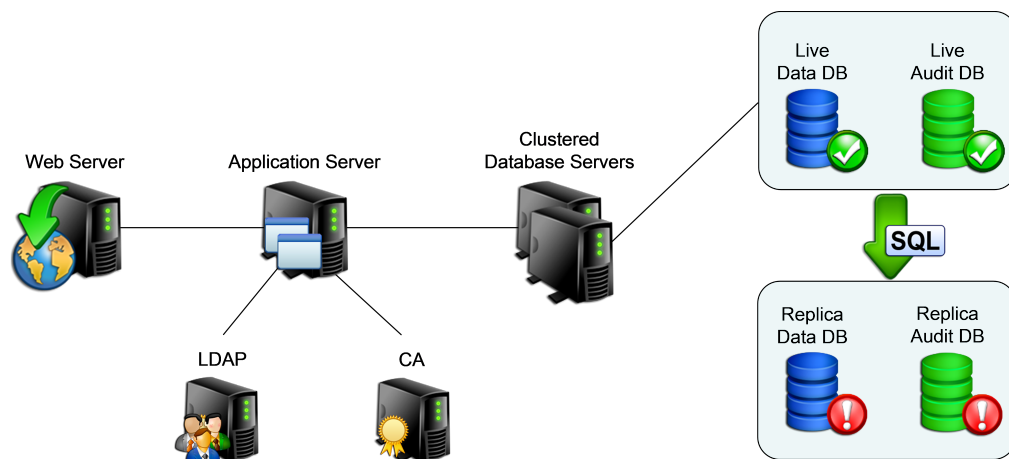
In the example above Microsoft SQL Server Clustering was used to provide failover only, no load balancing was used. In addition the audit database was split out from the main operational database.

The data storage requirements can be broken down into discrete components:

- Basic configuration data and so on is approximately 2MB, as this remains static regardless of the number of users stored, this can be ignored for storage calculations.

- User records

  - A typical user record with 2 fingerprints and a facial biometric averages 56KB.

    Note: If you are capturing extensive user biometric data such as 10 slap/roll prints, facial biometrics and EFT data, you require a significant amount of disk space to store this information on the database server.

    For example, a fully-enrolled person with 10 slap/roll prints, EFT and facial biometrics may require over 2MB of data.

  - From MyID 10.0, image data relating to user records is stored in the MyID database. This offers greater security, performance and backup options over storing images as files on the server. However, this also increases the size of the database. This is a configurable feature, so the range of data stored can cover photographs, uploaded images and scanned documents, but can be extended to add other uploaded document types such as PDFs or Word documents.

    The image stored in the database will be approximately the same size as stored on the file server.

- Each card record averages 2KB

- Each certificate record averages 13KB (CA dependent)

- Each audit record averages 2KB (variable dependent upon configuration)

Since audit information can be archived at configurable intervals, these figures are separated.

For a static deployment of 100,000 users with 100,000 cards and 200,000 certificates a storage requirements calculation would be as follows:

Note: For the purposes of this example, it is assumed that each user has 1 photograph (of 250KB) and 1 scanned document (of 250KB).

- User data = 100,000 users x 56KB = 5.34 GB

- User images and scanned documents = 100,000 users x 500KB = 48 GB

Note: The above estimate excludes any data stored outside of the database (for example, bureau export files). They also exclude the storage requirement of the database infrastructure itself (for example, transaction logs and indices).

9   Running multiple servers

A MyID system comprises a web server, an application server, and a database server. You may want to increase the number of servers of each type to provide more processing power, to distribute traffic, or to provide failover capability.

Note: Each web server must be paired with a single application server. You cannot load-balance the web server to application server traffic. However, you can have multiple web servers connected to the same application server – you can balance the client load across multiple web servers.

Where high availability is critical, you are advised to set up SQL Server failover clustering for your MyID database server. See your Microsoft documentation for details.

9.1   Multiple web servers

You can install multiple MyID web servers (and web services servers) that you can use in conjunction with a load balancer to distribute the network traffic across several servers.

Run the MyID application installation program on each web server and set up the COM proxies. See the *Split deployment* section of the ***Installation and Configuration Guide*** for details of installing the web server on a separate physical machine.

Once you have installed the web servers, configure your load balancer to distribute the traffic amongst the servers and set up session affinity. See your load balancer documentation for details.

See also the *Reverse proxies and load balancing* section in the ***Web Service Architecture*** guide.

9.1.1   Restricting available workflows

You can configure the web services to prevent clients from being able to view particular workflows. This is a global setting that affects all clients, unlike configuring the roles within MyID.

You may want to do this, for example, if you have multiple web servers operating in environments with different levels of security.

For more information, contact customer support, quoting reference SUP-256.

9.2   Multiple application servers

You can install multiple MyID application servers to work in conjunction with your multiple MyID web servers.

You can use your load balancer to distribute traffic to the different web servers, and then you can configure each web server to communicate with a different MyID application server. All the application servers are connected to the same MyID database.

When you install the COM proxies on the web servers, you can decide which application server to use; this allows you to distribute the load. For example, you might have four web servers and two application servers – web servers A and B have the proxies for application server Alpha, while web servers C and D have the proxies for application server Beta installed.

To set up multiple application servers:

1. Establish an operational MyID system using a single application server.

2. On the primary application server, export the registry key that contains the master key.

   The master key is located in the following part of the registry:

   `HKEY_LOCAL_MACHINE\SOFTWARE\Intercede\Edefice\Mastercard`

   You must make sure that all of the application servers use the same master key.

   If you are using an HSM, you must install the HSM client software before you can import the key onto the additional application server. You must follow the instructions in your HSM integration guide; however, you do not need to create a partition or run GenMaster, as these have already been carried out on the primary application server.

   You can use the SetHSMPIN utility to add the HSM PIN to the registry of the additional application server. See the *Setting the HSM PIN* section in the ***Installation and Configuration Guide*** for details.

   If you require additional information on using multiple application servers with HSMs, contact customer support, quoting reference SUP-90.

3. If you are using a Microsoft Windows CA, issue a new Enrollment Agent certificate along with its private key. You must also export the KRA certificate on the app server and import it to each application server.

4. On each additional application server:

   a. Import the master key registry settings.

   b. Run the MyID product installation program to install the application server.

   c. If you are using a Microsoft Windows CA, each additional application server requires an Enrollment Agent certificate.

      Normally this will be a different enrollment agent certificate for each application server, but if required you can export a copy of the enrollment agent certificate and private key from the original application server and configure on each additional application server.

      If you manually import the same enrollment agent certificate onto additional application servers, you must write the certificate to a certificate store called `edefice`, using the certutil utility:

      `certutil -addstore -f -user edefice my.cer`

      where `my.cer` is the name of the file to which you exported the certificate.

      Note: If your system uses a different certificate store for EA certificates, change `edefice` to the name of the appropriate store.

      Note: The private key must also be present on the machine; for example, imported as a pfx file.

   d. Disable the following service:

      - `eBureauService` (only installed on systems that have been configured for bureau integration)

This service must be running on the primary application server only. If you do not disable the service on the additional application servers, you may experience problems.

e. If you have multiple MyID certificate services, make sure you set the `RecordSize` parameter in the registry for each to a value of `1`.

The default registry location for this parameter is:

```
HKEY_LOCAL_
MACHINE\SYSTEM\CurrentControlSet\Services\eCertificateSrv\Parameters
```

Note: If you have multiple instances of the MyID certificate services running on the same application server, the registry key will be different from `eCertificateSrv` for the additional instances.

f. Export the COM proxies from the application server to the appropriate web server.

This allows you to distribute the traffic amongst your application servers.

g. On systems that use signing certificates (for example, PIV or CIV implementations):

i. Check the registry key on the primary application server:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Intercede\Edefice\PIV
```

ii. For each of the signing certificates (for example, `CHUIDSigningCertificate` or `SecurityObjectSigningCertificate`) check the location of the certificate, then copy the file from that location on the primary application server to the same location on the additional application server.

iii. Update the registry on the additional application server for each signing certificate to match the primary application server.

h. Confirm that the MyID `.udl` files in the Windows `System32` folder point to the correct database server.

These files start with the name you provided for the MyID database; for example, `MyID.udl`, `MyIDaudit.udl` and `MyIDarchive.udl`.

Note: To edit the `.udl` files, you must open a Windows command prompt, navigate to the `System32` folder, then type the name of the `.udl` file and press Enter.

i. Restart the MyID application server.

- **IKB-206 – Multiple email notifications**

If you have multiple certificate servers configured in your MyID system, whether on multiple application servers or using multiple instances of the certificate service on the same server, users may receive duplicate email notifications for certificate renewals. This is due to conflicting settings between the certificate services.

You are recommended to use an alternative configuration that prevents this situation by allowing the database server to control sending email notifications instead of the certificate services.

For more information, contact customer support, quoting reference SUP-253.

9.3 Multiple servers with a web server in a DMZ

You may want to configure your system to have a web server in a separate domain. This web server can serve as a bridge between the outside world and the protected network that hosts the MyID application server and database.

To configure this, you must set up local users on your application and web servers, and configure the firewall to allow communication between the servers.

1. On both the application server and web server in the separate domain, create the following user accounts, and add them to the Distributed COM Users group:

   • LocalWeb – this is the local account for running the MyID website.

   • LocalMWS – this is the local account for running the MyID web services.

   Note: These are suggested names for the local accounts. You can use your own names, as long as you use them consistently across both servers.

2. On the application server, in **Component Services**, under **My Computer > COM+ Applications**, expand the tree for the MyID component to view the **Roles** option, then add the appropriate LocalWeb and LocalMWS users to the MyID roles for each MyID component – these roles are:

   • App_Role

   • Web_Role

   Add the appropriate local users to each role that contains the existing domain users. That is, if the role has the MyID web domain user, add the local web user; if the role has the MyID web service domain user, add the local MWS user.

3. On the web server, install the MyID **Web Server** using the main MyID installation program.

   When you install the web server and web service components, specify the local users you created above. Specify the local machine name with the user name; for example:

   ```
   MYSERVER01\LocalWeb
   ```

4. Set up the Windows firewall between the web and application servers to allow the following:

   • 135/TCP – RPC Endpoint Manager.

   • 5000-5099/TCP – DCOM.

   • 49152-65535/TCP – RPC for LSA, SAM, Netlogon.

5. On the application server, copy the proxy MSI files for each of the MyID components.

   The proxy MSI files are located by default in the following folder:

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

6. Copy the proxy MSI files to the web server in the separate domain and install them.

9.3.1 Known issues

- **IKB-355 – You must enter the MyID COM+ user details even on a web-only tier**

  Currently the MyID installation program requires you to enter the COM+ account in this scenario. While this information is not technically required for operation, the installer currently needs this information to continue with the installation process.

10 Windows services

SIU references: SIU-252, SIU-253, SIU-254, SIU-255, SIU-256, SIU-257.

Note: If you install additional modules on your MyID system, you may see additional services running under the MyID user accounts. See the documentation provided with the additional modules for details.

If your system uses servers for multiple roles (for example, a combined application, web, and web services server, or an application server with a separate web/web services server) MyID will install the services for each role on the same server; however, the services will still run under their respective user accounts.

10.1 Application server services

The following is a list of the Windows services installed on the application server by MyID:

Name	Display Name	Startup	Description
eBureauSrv	eBureauSrv Services Server	Automatic	eBureauSrv Services Polling Agent. **Note:** Only present on systems that have been configured for bureau integration.
eCertificateSrv	eCertificate Services Server	Automatic (Delayed Start)	eCertificate Services Polling Agent.
eJobServer	eJobServer	Automatic	Asynchronous processing of CMS jobs.
eKeySrv	eKeyServer	Manual	Encryption Service – must be set to manual start.
eMessageSrv	eMessageServer	Manual	Messaging service.
MyIDExpiringItemsApp	MyID Expiring Items: App	Automatic (Delayed Start)	Used to monitor the MyID COM+ user account along with any certificates assigned to it. See section 3.1.1, *The monitoring services*
NotificationsService	MyID Notifications Service	Automatic	MyID Notifications service.
MyID SNMP Agent	MyID SNMP Agent	Automatic	SNMP agent used for monitoring. See section 3.3.5, *SNMP Agent notifications*

These services run using the MyID COM+ user.

10.2   Web server services

The following Windows service is installed on the web server by MyID:

Name	Display Name	Startup	Description
MyIDExpiringItemsWeb	MyID Expiring Items: Web	Automatic (Delayed Start)	Used to monitor the MyID IIS user account along with any certificates assigned to it. See section *3.1.1*, *The monitoring services*.

This service runs using the MyID IIS user.

10.3   Web services server services

The following Windows service is installed on the web services server by MyID:

Name	Display Name	Startup	Description
MyIDExpiringItemsMws	MyID Expiring Items: Mws	Automatic (Delayed Start)	Used to monitor the MyID web services user account along with any certificates assigned to it. See section *3.1.1*, *The monitoring services*.

This service runs using the MyID web services user.

11 Communication, security, and trust

The following descriptions explain the communication channels that are needed when deploying each server on a separate machine.

Note: Each Windows server used with MyID must be in the same, or a trusted, Windows domain. MyID relies on Windows domain level security.

If you need to work in a non-domain environment, or require additional levels of security between servers, contact customer support quoting reference SUP-74 to discuss your requirements.

11.1 Client to web server

MyID client to web server communication is performed via HTTP or HTTPS (recommended).

HTTPS can be used in either server-authentication only mode, or to authenticate both client and server. This is achieved through standard Windows IIS configuration. For production environments, as a minimum, you must set up one-way SSL/TLS to ensure encryption of traffic between the client and the web server.

MyID web-based clients require some ActiveX components in order to allow communication between the browser and other devices such as smart card readers, biometric devices, card printers, cameras, or scanners.

These components may be installed locally from installation media, or downloaded as an Authenticode-signed cabinet from the MyID server, or pushed out as an msi file using group policy.

11.2 Web server to application server

The web server comprises ASP, HTML, JS, XSL, XML and image files, as well as ASP.NET and WCF web services. These communicate with the MyID middleware components using COM+ object instantiation.

- Where the middleware is co-located on the web server, this communication is directly through the COM+ framework.

- If the middleware is located on a separate MyID server, this communication is performed through the DCOM architecture.

DCOM uses RPC running over a TCP protocol which, by default, assigns communications ports on demand in the range 1024 to 65535. It is possible to restrict the ports used – this is important if you intended placing a firewall between the website and the MyID server. You will also need to enable port 135 to support the 'end point mapper'. The Windows authentication and Active Directory protocols must also be opened. For further details see the document entitled "How to Configure Firewalls for Domains and Trusts" on the Microsoft website.

Note: DCOM will operate through a firewall *provided that it does not perform network address translation*. For further details see the document entitled "Using Distributed COM with Firewalls" on the Microsoft website.

Before installing MyID, verify that the necessary bidirectional RPC communication is available by using the Microsoft **DTCPing** tool (available from the Microsoft website).

In addition to the above, it is recommended that additional intrusion detection software is implemented on the web server to prevent security breaches through unauthorized changes to the website.

11.2.1   DCOM port ranges

To force the RPC system to use a specific range for its dynamic ports:

1. From the Windows **Administrative Tools**, select **Component Services**.

2. Browse to **Console Root > Component Services > Computers**.

3. Right-click **My Computer** and select **Properties**.

4. Select the **Default Protocols** tab, ensure **Connection-oriented TCP/IP** is selected in the list and click the **Properties** button.

5. Set a port range.

   You should ensure the base port is above 1024. You need a range of at least 100 ports; for example, `5000-5099`.

6. Add the range, then click **OK**.

The port limit is not active until you reboot; however, you should set up the firewall before you reboot the machine.

11.2.2   Firewall configuration

You must open ports for the following:

- The ports to/from the Domain Controller to allow the web user to be authenticated.

- The DCOM port range you have set up (for example, 5000-5099).

- The RPC port (135). There is a predefined rule for port 135 called **COM+ Network Access** that you can enable.

- The HTTP or HTTPS ports (for example, 80 or 443). You need to open these ports from the application server to the web server, but not from the web server to the application server. General communications between the web server and the application server are carried out purely by DCOM – however, if you are using specific services on the web server that the application server needs to access (for example, for notifications, bureau, PACS, or web-hosted uploaded images) you must open the HTTP or HTTPS port.

- The ports for any external systems with which MyID needs to communicate.

See the documentation for the firewall you are using to open the necessary range of ports.

For example, to set up the default Windows firewall to use ports 5000-5099:

1. From the Windows **Administrative Tools**, select **Windows Firewall with Advanced Security**.

2. Select **Inbound Rules** and add a new rule using the **Actions** on the right.

3. In the wizard that appears, select **Port** for the rule type and click **Next**.

4. Select **TCP**.

5. Provide a list of the ports you specified in **Component Services**.

   You can specify a range; for example:

   `5000-5099`

6. Click **Next**.

7. Select **Allow the Connection** then click **Next**.

8. Make sure all three **Apply** rules are selected then click **Next**.

9. Type a name for the rule.

10. Finish the wizard.

11. Ensure the firewall is switched on, then reboot the machine

    Note: You must carry out this procedure on both the web server and the application server.

11.3 Application server to database server

SIU references: SIU-226, SIU-227, SIU-228, SIU-229, SIU-230, SIU-231, SIU-232, SIU-261, SIU-262, SIU-317.

The MyID application server comprises middleware components that implement business object logic, interact with the various additional services (CA, authentication, and so on) and abstracts the storage and retrieval of persistent data. This demands a connection to the database.

Database connectivity is achieved using data link (UDL) files that communicate through DTC/RPC. As it is possible to split the main, audit, archive, and binary image databases across separate data sources, the following data link files are created. These may all point to the same database or different databases. In addition, a data link is created for the authentication database, which is always a separate database. The data link files are created in the `System32` directory of the `Windows` directory with the following names:

- `<databasename>.udl`

- `<databasename>audit.udl`

- `<databasename>archive.udl`

- `<databasename>auth.udl`

- `<databasename>binary.udl`

The MyID COM+ account must be given at least read rights to the `.udl` files subsequently used to access the database server.

MS DTC must be configured to allow network DTC access on both the database server and the MyID server. See the *MSDTC security configuration* section in the *Installation and Configuration Guide* for details.

It is also recommended that the SQL Server network utility on the database server and the SQL client network utility on the MyID server are configured to use TCP/IP only (*not* Named Pipes). You are recommended to disable Named Pipes on the application and database servers in **SQL Native Client 11.0 Configuration (32bit)** and **SQL Server Network Configuration** in the SQL Server Configuration Manager – if you have a firewall configured between the application server and database server, this step is essential.

It is recommended that the Microsoft **DTCTester** tool is used to confirm connectivity. This is available from the Microsoft website.

11.3.1 DCOM port ranges

You must set up a range of approximately 100 ports to use between the application server and the database server.

11.3.2 Firewall configuration

You must open ports for the following:

- The DCOM port range you have set up. You must open the firewall for these ports in both directions.
- The SQL Server port (by default, 1433). Set up the firewall to allow communication from the SQL Server port to ANY, and from ANY to the SQL Server port.

See the documentation for the firewall you are using to open the necessary range of ports.

11.3.3 Encrypting the connection to SQL Server

The MyID application server communicates with the database server using the data link (UDL) files described above – typically this is configured to use Microsoft OLE DB Driver for SQL Server which in turn results in the database communications using MS DTC/RPC. By default these calls are made unencrypted.

For deployments where the MyID application server and database server reside in a dedicated secure server environment, this unencrypted transmission of data from server to server is not typically seen as a risk. That is, the physical and firewall separation of the server environment is enough to satisfy any security concerns.

However, for deployments where an extra level of security is required for communicating with the SQL Server database server, configuring SQL Server to use SSL/TLS will ensure that all data transmitted from MyID to the database is encrypted.

This is a SQL Server configuration rather than a MyID configuration. Instructions are provided by Microsoft in Technet article ms189067.

The following points should be noted if you want to set this configuration:

- SSL/TLS is a server-wide setting in SQL Server so enabling this configuration will affect every hosted database.
- There will potentially be a minor performance impact due to the SSL/TLS handshaking and encryption/decryption overheads.

12 Other considerations

12.1 Application pools

SIU references: SIU-233, SIU-234.

MyID uses the following application pools:

- **MyIDPoolClassic** – uses a **Managed Pipeline** of **Classic**. Used for the MyID websites (including each language variant – **MyID\en** and **MyID\us** for example).
- **MyIDWebService** – uses a **Managed Pipeline** of **Integrated**. Used for the MyID web services (MyIDDataSource, MyIDProcessDriver, MyIDEnroll).

The application pools are created by the MyID product installation program.

12.2 Operating across multiple time zones

MyID can be deployed in environments where the servers and clients span time zones. The MyID database server time is treated as the definitive time source, so it is important that this is synchronized with the domain controller time, which should ideally be synchronized with a trusted time source.

Dates and times are stored in the database using UTC.

Clients connecting to MyID will operate in their own locales, but all date and time information transmitted back to the servers is first converted to UTC. Individual records (card expiry dates and so on) are converted back to local time on retrieval, but it should be noted that the audit records are always reported using the database local server time, so that consistent comparisons can be made between the data seen by the users and central administrators. Events in the **System Events** workflow are always displayed in UTC.

12.3 Running multilingual environments

MyID can operate multiple websites, each with its own translated version of the software. MyID automatically detects the locale of the client and route the connection to the appropriate translated version. For details of how to configure multilingual sites contact customer support quoting reference SUP-138.