MyID

Version 11.7

Configuring Logging

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

Copyright

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

Licenses and Trademarks

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 and to indicate references to other sections within the current document:

  For example:

  - "Copy the file *before* starting the installation"
  - "See *Issuing a Card* for further information"

- ***Bold and italic*** 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 media.

- A `fixed width` font is used where the identification of spaces is important, including filenames, example SQL queries and any entries made directly into configuration files or the database.

- **Notes** are used to provide further information, including any prerequisites or configuration additional to the standard specifications.

  For example:

  Note: This issue only occurs if updating from a previous version.

- Warnings are used to indicate where failure to follow a particular instruction may result in either loss of data or the need to manually configure elements of the system.

  For example:

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

Contents

1 Introduction

This document describes how to set up logging for various MyID® systems, including:

- MyID Desktop
- MyID Self-Service App
- MyID Self-Service Kiosk
- MyID Image Capture
- MyID Windows Integration Service (WSVC)
- MyID Identity Agent
- MyID Client Components
- MyID Web Service Architecture
- MyID REST and Authentication Web Services
- Other MyID web services
- MyID server components

Important: Use this document only in conjunction with advice from customer support. Log files are not intended to be readable by customers, and may require expert analysis. Do not leave logging switched on when you do not need to; the files may become very large and may impact performance. Log files may also contain sensitive information. Always back up your system before making any changes; switching on logging may require the manual editing of configuration files or the system registry.

1.1 Change history

Version	Description
`INF1875-01`	First release.
`INF1875-02`	Renamed from ***Windows Client Logging*** to ***Configuring Logging***. Updated to include logging for other systems.
`INF1875-03`	The Self-Service App no longer requires a separate .exe for automation mode.
`INF1875-04`	Added information about logging the MyID Rest and authentication web services, which are used for the MyID Operator Client.
`INF1875-05`	Added to the main documentation set. Added information about logging for the `WHfB` (Windows Hello for Business) client component.

2 Windows Clients

2.1 Logging for MyID Desktop

You can set up your MyID Desktop application to write debug information to a log file. You may need to provide this information to Intercede customer support.

To set up the logging, you must create a text file in the application folder with the relevant information.

To switch logging on:

1. On the client PC, if it does not already exist, create a `Debug.config` file in the following folder:

   `C:\Program Files (x86)\Intercede\MyIDDesktop\`
2. Using a text editor, open the `Debug.config` file.

   Note: Make the changes to the config file exactly as shown. The case is important.

```
<Debug>
  <add key="Enabled"                          value="true" />
  <add key="DebugProvider"                    value="CodeHelpers.DebugTrace" />
  <add key="LogFirstChangeExceptions"         value="true" />
  <add key="LogUnhandledExceptions"           value="false" />
  <add key="LogThreadingExceptions"           value="false" />
</Debug>
```

3. You can switch logging on and off by editing the following:
   - `Enabled` – set to `true` to enable logging, or `false` to disable logging.
4. Save the configuration file.
5. On the client PC, back up the `MyIDDesktop.exe.config` file in the following folder:

   `C:\Program Files (x86)\Intercede\MyIDDesktop\`
6. Using a text editor, open the `MyIDDesktop.exe.config` file.

   Note: Make the changes to the config file exactly as shown. The case is important.
7. Within the root <`configuration`> node of the file, add the following immediately before the line containing `</configuration>`:

```
<log4net>
 <root>
 <level value="ALL" />
 <appender-ref ref="RollingFileAppender" />
 </root>
 <appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
 <file type="log4net.Util.PatternString" value="LogFile.xml"/>
 <appendToFile value="true"/>
 <datePattern value="yyyyMMdd"/>
 <rollingStyle value="Size"/>
 <maxSizeRollBackups value="10"/>
 <maximumFileSize value="10000KB"/>
 <layout type="log4net.Layout.XmlLayoutSchemaLog4j">
 <locationInfo value="true"/>
 </layout>
 </appender>
</log4net>
<Debug configSource="Debug.Config" />
```

Note: The following line contains the name of the file to which the log is written:

```
<file type="log4net.Util.PatternString" value="LogFile.xml"/>
```

If you do not specify a path, the file is written to the program folder. Make sure that the user running MyID Desktop has permission to write to this folder. If the user does not have permission to write to the program folder, specify a path to a file that the user *does* have permissions to; for example:

```
<file type="log4net.Util.PatternString" value=" C:\Logs\log.xml"/>
```

8. Save the config file.

2.2 Logging for the Self-Service App

You can set up your Self-Service App to write debug information to a log file. You may need to provide this information to Intercede customer support.

To set up the logging, you must create a text file in the application folder with the relevant information.

To switch logging on:

1. On the client PC, if it does not already exist, create a `Debug.config` file in the following folder:

   `C:\Program Files (x86)\Intercede\MyIDApp\Self Service Application\`

2. Using a text editor, open the `Debug.config` file.

   Note: Make the changes to the config file exactly as shown. The case is important.

```
<Debug>
  <add key="Enabled"                        value="true" />
  <add key="DebugProvider"                  value="CodeHelpers.DebugTrace" />
  <add key="LogFirstChangeExceptions"       value="true" />
  <add key="LogUnhandledExceptions"         value="false" />
  <add key="LogThreadingExceptions"         value="false" />
</Debug>
```

3. You can switch logging on and off by editing the following:

   - `Enabled` – set to `true` to enable logging, or `false` to disable logging.

4. Save the configuration file.
5. On the client PC, back up the `MyIDApp.exe.config` file in the following folder:

   `C:\Program Files (x86)\Intercede\MyIDApp\Self Service Application\`

6. Using a text editor, open the `MyIDApp.exe.config` file.

   Note: Make the changes to the config file exactly as shown. The case is important.

7. Within the root `<configuration>` node of the file, add the following immediately before the line containing `</configuration>`:

```
<log4net>
 <root>
 <level value="ALL" />
 <appender-ref ref="RollingFileAppender" />
 </root>
 <appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
 <file type="log4net.Util.PatternString" value="LogFile.xml"/>
 <appendToFile value="true"/>
 <datePattern value="yyyyMMdd"/>
 <rollingStyle value="Size"/>
 <maxSizeRollBackups value="10"/>
 <maximumFileSize value="10000KB"/>
 <layout type="log4net.Layout.XmlLayoutSchemaLog4j">
 <locationInfo value="true"/>
 </layout>
 </appender>
</log4net>
<Debug configSource="Debug.Config" />
```

Note: The following line contains the name of the file to which the log is written:

```
<file type="log4net.Util.PatternString" value="LogFile.xml"/>
```

8. Save the config file.

2.3 Logging for the Self-Service Kiosk

You can set up your Self-Service Kiosk to write debug information to a log file. You may need to provide this information to Intercede customer support.

To set up the logging, you must create a text file in the application folder with the relevant information.

To switch logging on:

1. On the client PC, if it does not already exist, create a `Debug.config` file in the following folder:

   `C:\Program Files (x86)\Intercede\MyIDSelfServiceKiosk\`

2. Using a text editor, open the `Debug.config` file.

   Note: Make the changes to the config file exactly as shown. The case is important.

```
<Debug>
  <add key="Enabled"                        value="true" />
  <add key="DebugProvider"                  value="CodeHelpers.DebugTrace" />
  <add key="LogFirstChangeExceptions"       value="true" />
  <add key="LogUnhandledExceptions"         value="false" />
  <add key="LogThreadingExceptions"         value="false" />
</Debug>
```

3. You can switch logging on and off by editing the following:
   - `Enabled` – set to `true` to enable logging, or `false` to disable logging.
4. Save the configuration file.
5. On the client PC, back up the `MyIDKiosk.exe.config` file in the following folder:

   `C:\Program Files (x86)\Intercede\MyIDSelfServiceKiosk\`

6. Using a text editor, open the `MyIDKiosk.exe.config` file.

   Note: Make the changes to the config file exactly as shown. The case is important.

7. Within the root `<configuration>` node of the file, add the following immediately before the line containing `</configuration>`:

```
<log4net>
 <root>
 <level value="ALL" />
 <appender-ref ref="RollingFileAppender" />
 </root>
 <appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
 <file type="log4net.Util.PatternString" value="LogFile.xml"/>
 <appendToFile value="true"/>
 <datePattern value="yyyyMMdd"/>
 <rollingStyle value="Size"/>
 <maxSizeRollBackups value="10"/>
 <maximumFileSize value="10000KB"/>
 <layout type="log4net.Layout.XmlLayoutSchemaLog4j">
 <locationInfo value="true"/>
 </layout>
 </appender>
</log4net>
<Debug configSource="Debug.Config" />
```

Note: The following line contains the name of the file to which the log is written:

```
<file type="log4net.Util.PatternString" value="LogFile.xml"/>
```

8. Save the config file.

3 MyID Image Capture

To set up logging, create a text file called `LogConfig.xml` and add the following to it:

```
<log4net>
       <root>
              <level value="ALL" />
              <appender-ref ref="RollingFileAppender" />
       </root>
       <appender name="RollingFileAppender"
type="log4net.Appender.RollingFileAppender">
              <file type="log4net.Util.PatternString" value="%property{LogFile}"/>
              <appendToFile value="true"/>
              <datePattern value="yyyyMMdd"/>
              <rollingStyle value="Size"/>
              <maxSizeRollBackups value="10"/>
              <maximumFileSize value="10000KB"/>
              <layout type="log4net.Layout.XmlLayoutSchemaLog4j">
                     <locationInfo value="true"/>
              </layout>
       </appender>
</log4net>
```

Copy this file into the following folder on the client PC:

`%UserProfile%\AppData\LocalLow\Intercede\ImageCapture\`

Note: This folder is created the first time you run the Image Capture component.

The Image Capture component creates the following log files:

- `MyIdImageCaptureActiveXLog.xml`

  This corresponds to the ActiveX control with which Internet Explorer is invoking and communicating.

- `MyIdImageCaptureComLog.xml`

  This corresponds to the COM object that the ActiveX control invokes to launch the Image Capture control in its own process.

- `MyIdImageCaptureLog.xml`

  This corresponds to the WPF control itself, which is invoked by the COM object.

4 MyID Windows Integration Service (WSVC)

To set up logging, create a text file called `LogConfig.xml` and add the following to it:

```
<log4net>
       <root>
              <level value="ALL" />
              <appender-ref ref="RollingFileAppender" />
       </root>
       <appender name="RollingFileAppender"
type="log4net.Appender.RollingFileAppender">
              <file type="log4net.Util.PatternString" value="%property{LogFile}"/>
              <appendToFile value="true"/>
              <datePattern value="yyyyMMdd"/>
              <rollingStyle value="Size"/>
              <maxSizeRollBackups value="10"/>
              <maximumFileSize value="10000KB"/>
              <layout type="log4net.Layout.XmlLayoutSchemaLog4j">
                     <locationInfo value="true"/>
              </layout>
       </appender>
</log4net>
```

Copy this file into the following folder on the client PC:

`C:\Program Files (x86)\Intercede\MyID_Client_Service\`

5 MyID Identity Agent

You can configure the Identity Agent app to create a log file for debugging purposes. Customer support may ask you to set the log level and send the resulting log file to Intercede for analysis.

Note: The Identity Agent app uses the system default email app to send the log file. For iOS devices, this means that you must have Apple Mail configured with at least one email account.

To enable logging, use the following configuration options on the **Identity Agent Policy** page of the **Operation Settings** workflow:

- **Administrator email address** – Set this to the email address to which Identity Agent will send logs for troubleshooting purposes.
- **Log level** – Set this to the level of debug logging you want Identity Agent to produce. Higher levels result in more detail, but larger files.

  Set to one of the following:

  `0 – NONE`

  `1 – FATAL`

  `2 – ERROR`

  `3 – WARNING`

  `4 – INFO`

  `5 – DEBUG`

  `6 – VERBOSE`

  By default, the log level is set to level 2, `ERROR`.

  Note: This setting affects the level of *debug* logging only; the Identity Agent also logs all *messages* that occur between the client and the server. If you want to switch off logging altogether, set the **Maximum number of log files** to `0`.

- **Maximum log storage space** – The maximum amount of space (in MB) that log files will take up on a device. Once this limit is reached, log files will be deleted automatically, oldest first, to clear room for new files.
- **Maximum number of log files** – The maximum number of log files to be stored on a device. Once this limit is reached, log files will be deleted automatically, oldest first, to clear room for new files.

  To allow as many files as will fit in the maximum log storage space, set this value to `-1`. This is the default setting.

  To switch off logging, set this value to `0`.

6 MyID Client Components

The MyID Client Components provide logging for a variety of the components in the UMC package.

You can set up logging for the following components individually:

- `ApduScript`
- `CanonCapture`
- `ClientVersion`
- `CSP COM`
- `CSPCertEnroll`
- `DataExchange`
- `DirectAPISmartCard`
- `ECardPrintX`
- `Edefice_OCR`
- `EdeficeSmartCard`
- `Envelope COM`
- `eSCardCOM`
- `FileUtils`
- `MifareCom`
- `ScannerCapture`
- `SmartcardKeypair`
- `WHfB`

To set up logging for a component:

1. Set the following in the client PC's or application server's registry:

   `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Intercede\Edefice\Trace`

   If the `Trace` key does not exist, you must create it.
2. In the `Trace` key, create a DWORD value with the name of the component from the list above; for example, `EdeficeSmartCard`. Set the value to `1` to enable logging, and `0` to disable logging.

   Note: For the `WHfB` (Windows Hello for Business) component, you must set the value to `9` to enable logging, as this component supports only parameter-level tracing.
3. In the `Trace` key, create a key with the name of the component; for example, `EdeficeSmartCard`. Within this key, create a string value called `Location` and set this to the full path of the file to which you want to send the log information.

Note: If you are on the server, you must ensure that the MyID named COM user has the necessary permissions to create and write to the log file. You can create a file then give the user write permissions if you prefer not to give the user create permissions.

Note: You must ensure that all users can write to the location; set the permissions of this folder to be **Everyone - Full control**.

Important: Disable the logging when you have completed diagnosing the issues, as the log file may become very large.

7 MyID Client Service

The MyID Client Service is installed on client PCs and provides access to smart cards and other hardware component for the browser-based MyID Operator Client.

To enable logging in the MyID Client Service, create a new file called `Log.config` in the same directory as `MyIDClientService.exe` and add the following content:

```
<configuration>
  <configSections>
    <section name="log4net"
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
  </configSections>
  <log4net>
    <appender name="RollingFileAppender"
type="log4net.Appender.RollingFileAppender">
      <file value="C:\Logs\MCS.log" /> <!-- Set this to the path you want the log-
file to be saved to -->
      <appendToFile value="true" />
      <rollingStyle value="Size" />
      <maxSizeRollBackups value="5" />
      <maximumFileSize value="10MB" />
      <staticLogFileName value="true" />
      <layout type="Intercede.MyID.Logging.Log4Net.LogLayout,
Intercede.MyID.Logging.Log4Net" />
    </appender>
    <root>
      <level value="All" />
      <appender-ref ref="RollingFileAppender" />
    </root>
  </log4net>
</configuration>
```

Set the `value` of the `file` node to the path and name of the file to which you want to write the log. The logged-on user must have read-write access to this file.

To disable logging, delete the `Log.config` file.

8 MyID Web Services

You can set up logging for the following web services:

- `MyIDDataSource`
- `MyIDProcessDriver`

To set up logging:

1. In a text editor, open the `Log.config` file for the component you want to log.

   For `MyIDDataSource`, this is:

   ```
   C:\Program Files (x86)\Intercede\MyID\SSP\MyIDDataSource\Log.config
   ```

   For `MyIDProcessDriver`, this is:

   ```
   C:\Program Files (x86)\Intercede\MyID\SSP\MyIDProcessDriver\
   Log.config
   ```

2. Set the value of the `file` node to the output location; for example:

   ```
   <file value="C:\logs\MyIDDataSource.log" />
   ```

3. Replace the following line:

   ```
   <level value="OFF" />
   ```

   with:

   ```
   <level value="All" />
   ```

4. Save the file.

Note: You must ensure that the MyID web service user has the necessary permissions to create and write to the log file. You can create a file then give the user write permissions if you prefer not to give the user create permissions.

The log is set to a maximum of 60MB, split over six rolling files.

Important: The log files may contain personal data, including names and addresses. Make sure you delete these logs as soon as possible.

9 MyID REST and Authentication Web Services

MyID provides web services for the MyID Operator Client to communicate with and authenticate to the web server.

You can set up logging for the following web services:

- `rest.core`
- `web.oauth2`

To set up logging:

1. In a text editor, open the `Log.config` file for the web service you want to log.

   For `rest.core`, this is:

   ```
   C:\Program Files (x86)\Intercede\MyID\rest.core\Log.config
   ```

   For `web.oauth2`, this is:

   ```
   C:\Program Files (x86)\Intercede\MyID\web.oauth2\Log.config
   ```

2. Set the value of the `file` node to the output location; for example:

   ```
   <file value="C:\logs\rest.core.log" />
   ```

3. Edit the following line:

   ```
   <level value="OFF" />
   ```

   and replace the `OFF` value with one of the following:

   `ALL`

   `DEBUG`

   `INFO`

   `WARN`

   `ERROR`

   `FATAL`

   These error levels generate different levels of detail in the log, from most (`ALL`) to least (`FATAL`). To switch logging off altogether, set the value back to `OFF`.

   Important: Log levels `ALL` and `DEBUG` log all COM calls including parameters sent to and from the MyID application server. This produces a high volume of log information and may contain personal data. Reduce the log level, or set it to `OFF`, as soon as possible once you have obtained the relevant logging details.

4. Save the file.

Note: You must ensure that the MyID web service user has the necessary permissions to create and write to the log file. You can create a file then give the user write permissions if you prefer not to give the user create permissions.

The log is set to a maximum of 60MB, split over six rolling files.

Important: The log files may contain personal data, including names and addresses. Make sure you delete these logs as soon as possible.

9.1 Logging Microsoft components

The MyID REST and authentication web services rely on a Microsoft stack, and you can add extra logging for these components to provide information on issues deeper in the stack (for example, JWT validation failures, or ASP.net infrastructure issues).

To enable this logging:

1. In a text editor, open the `appsettings.json` file for the web service.

   For `rest.core`, this is:

   ```
   C:\Program Files (x86)\Intercede\MyID\rest.core\appsettings.json
   ```

   For `web.oauth2`, this is:

   ```
   C:\Program Files (x86)\Intercede\MyID\web.oauth2\appsettings.json
   ```

2. In the `Logging` section, add a new entry for Microsoft components.

   For example, if your file contains the following:

   ```
   "Logging":  {
        "LogLevel":  {
               "Default":  "Warning"
        }
   }
   ```

   Edit it to add an entry next to the `"Default":  "Warning"` as follows:

   ```
   "Logging":  {
        "LogLevel":  {
               "Default":  "Warning",
               "Microsoft": "Information"
        }
   }
   ```

   This example adds logging information from all Microsoft components at `Information` level.

   The supported log levels are different from the values in the `Log.config` file. From most detail to least, the options are:

   ```
   Trace
   Debug
   Information
   Warning
   Error
   Critical
   None
   ```

Note: The log level for `Default` in the `appsettings.json` file is ignored – this setting is controlled by the log level in the `Log.config` file. You must make sure that the `Log.config` file is configured to produce a log, or the additional Microsoft logging information will not be logged.

For more information, search for the *Logging in .NET Core and ASP.NET Core* article on the Microsoft website.

10 Other MyID Web Services

You can set up logging for the following web services:

- Credential Web Service (CWS)
- Device Management API (DWS)
- Lifecycle API (MyIDEnroll)

These web services use the same method of configuring logging.

Note: For each web service, you must ensure that the MyID web service user has the necessary permissions to create and write to the log file. You can create a file then give the user write permissions if you prefer not to give the user create permissions.

10.1 Credential Web Service

To set up logging for the web service, copy the following into a text file called `Log.Config` in the following folder:

`C:\Program Files (x86)\Intercede\MyID\SSP\CredentialWebService`

```
<configuration>
  <configSections>
    <section name="log4net"
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
  </configSections>
  <log4net>
    <appender name="RollingFileAppender"
type="log4net.Appender.RollingFileAppender">
      <file value="CredentialWebService.log" />
      <appendToFile value="true" />
      <rollingStyle value="Size" />
      <maxSizeRollBackups value="5" />
      <maximumFileSize value="10MB" />
      <staticLogFileName value="true" />
         <layout type="CredentialWebService.Web.MyXmlLayout" />
    </appender>
    <root>
      <level value="ALL" />
      <appender-ref ref="RollingFileAppender" />
    </root>
  </log4net>
</configuration>
```

10.2 Device Management API

To set up logging for the web service, copy the following into a text file called `Log.Config` in the following folder:

`C:\Program Files (x86)\Intercede\MyID\SSP\DeviceManagementAPI`

```
<configuration>
  <configSections>
    <section name="log4net"
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
  </configSections>
  <log4net>
    <!-- ConsoleAppender -->
    <appender name="ConsoleAppender" type="log4net.Appender.ConsoleAppender">
      <layout type="log4net.Layout.PatternLayout">
        <conversionPattern value="%-4timestamp [%thread] %-5level -
%message%newline%newline" />
      </layout>
    </appender>
    <!-- Rolling file appender to ProcessDriver.log-->
    <appender name="RollingFileAppender"
type="log4net.Appender.RollingFileAppender">
      <file value="DeviceManagementAPI.log" />
      <appendToFile value="true" />
      <rollingStyle value="Size" />
      <maxSizeRollBackups value="5" />
      <maximumFileSize value="10MB" />
      <staticLogFileName value="true" />
      <!--<layout type="log4net.Layout.PatternLayout">
        <conversionPattern value="%-4timestamp [%thread] %-5level -
%message%newline%newline" />
      </layout>-->
         <layout type="DeviceManagementAPI.MyXmlLayout" />
    </appender>
    <!-- Set root logger level to INFO and its only appender to A1 -->
    <root>
      <level value="ALL" />
      <!-- uncomment to see logging to output window -->
      <!-- <appender-ref ref="ConsoleAppender" />-->
      <appender-ref ref="RollingFileAppender" />
    </root>
  </log4net>
</configuration>
```

10.3 Lifecycle API

To set up logging for the web service, copy the following into a text file called `Log.Config` in the following folder:

`C:\Program Files (x86)\Intercede\MyID\Web\MyIDEnroll`

```
<configuration>
  <configSections>
    <section name="log4net"
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
  </configSections>
  <log4net>
    <appender name="RollingFileAppender"
type="log4net.Appender.RollingFileAppender">
      <layout type="MyIDEnroll.LogLayout" />
      <file value="MyIDEnroll.log" />
      <appendToFile value="true" />
      <rollingStyle value="Size" />
      <maxSizeRollBackups value="5" />
      <maximumFileSize value="10MB" />
      <staticLogFileName value="true" />
    </appender>
    <root>
      <level value="ALL" />
      <appender-ref ref="RollingFileAppender" />
    </root>
  </log4net>
</configuration>
```

11 Server Component Logging

You can set up logging for a variety of server components. The method for configuring logging depends on the component you want to log.

11.1 Registry method

You can use the registry method of configuring logging for the following components:

- `AccessProfileImport`
- `ADDeletionSync`
- `ADDeletionSync`
- `AdjudicationEquifax`
- `AdjudicationOPM`
- `AMAGPACSConnector`
- `ASyncImport`
- `BOL_Authentication`
- `BOL_Certificates`
- `BOL_Core`
- `BOL_DeviceManagement`
- `BOL_DevicePolicy`
- `BOL_Devices`
- `BOL_ImportFromCard`
- `BOL_Jobs`
- `BOL_LDAP`
- `BOL_Notifications`
- `BOL_People`
- `CardScriptExtensions`
- `CBPACSConnector`
- `CertificateRevocationConnector`
- `CertificateSrv`
- `eActivIDSDSProcessor`
- `eBureauSrv`
- `ECardPrintX`
- `eConfiguration`
- `eCS`
- `Edefice_CS`
- `Edefice_DAL`
- `EdeficeBOL_PKI`
- `EdeficeSmartCard`
- `eDirectory`
- `eEMVDataProcessor`

- eJobMaintenanceProcessor
- eJobServer
- eKeySrv
- eKeySrvPool
- Entrust_Admin
- EntrustJTKConnector
- ePkiConfig
- eStaleJobProcessor
- GEFCPACSConnector
- GEPACSConnector
- GPOBureauMessage
- HSMTestUtility
- ImportProcessor
- JobBatch
- LUNAKeySrv
- MicrosoftConnector
- MicrosoftKeyStore
- MifareCom
- NCKeySrv
- OfflineRevocationConnector
- OpenPlatformSecurity
- PivDataProcessor
- PivTransport
- PreciseConnector
- ResyncByCounter
- SecugenConnector
- SymantecLH
- SymantecMPKIHelper
- SunOne
- THNGHooks
- Unicert

You can also set up logging for any component that ends `BureauTransport`; for example, `GenBureauTransport`.

You can set up logging for the `Notifications` component, but this is the older component – for the current Notifications system, see section *11.2*, *Log4Net method*.

You can set up logging for the `eSCardCOM` component, but only after installing a debug version of the DLL. For example, for MyID PIV 9.0 SP1, the diagnostic patch D901MP316 is available.

To set up logging for a component:

1. Set the following in the MyID application server's registry:

   `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Intercede\Edefice\Trace`

   If the `Trace` key does not exist, you must create it.

2. In the `Trace` key, create a DWORD value with the name of the component from the list above; for example, `TPMManager`. Set the value to `1` to enable logging, and `0` to disable logging.

3. In the `Trace` key, create a key with the name of the component; for example, `TPMManager`. Within this key, create a string value called `Location` and set this to the full path of the file to which you want to send the log information.

Note: You must ensure that the MyID named COM user has the necessary permissions to create and write to the log file. You can create a file then give the user write permissions if you prefer not to give the user create permissions.

Important: Disable the logging when you have completed diagnosing the issues, as the log file may become very large.

11.1.1 Bureau logging

Logging for the Bureau server components is a variation on the standard registry method.

To set up logging for the bureau components:

1. Set the following in the MyID application server's registry:

   `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Intercede\Edefice\Trace`

   If the `Trace` key does not exist, you must create it.

2. In the `Trace` key, create the following keys:

   - `eBureauSrv`
   - `Boewe`

3. Inside each of the above keys, create a string value called `Logfile` and set this to the full path of the file to which you want to send the log information.

Note: You must ensure that the MyID named COM user has the necessary permissions to create and write to the log file. You can create a file then give the user write permissions if you prefer not to give the user create permissions.

Important: Disable the logging when you have completed diagnosing the issues, as the log file may become very large.

11.2 Log4Net method

You can use the Log4Net method of configuring logging for the following components:

- EJBCA connector
- SymantecMPKI connector
- Notifications.Net
- MyIDMailer

When you switch on logging, it generates log information for all of the above components. You cannot decide to log individual components.

To set up logging for these components, copy the following into a text file called `Log.Config`:

```
<?xml version="1.0" encoding="utf-8" ?>
 <configuration>
 <configSections>
 <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler,
log4net" />
 </configSections>
 <log4net>
 <appender name="MyIdLogFile" type="log4net.Appender.RollingFileAppender">
 <file value="c:\Logs\log.txt" />
 <appendToFile value="true" />
 <lockingModel type ="log4net.Appender.FileAppender+MinimalLock" />
 <maxSizeRollBackups value="10" />
 <maximumFileSize value="32Mb" />
 <rollingStyle value="Size" />
 <staticLogFileName value="true" />
 <layout type="log4net.Layout.PatternLayout">
 <header value="[Header] "/>
 <footer value="[Footer] "/>
 <conversionPattern value="%date [%thread] %-5level %logger - %message %newline" />
 </layout>
 </appender>
 <root>
 <level value="ALL" />
 <appender-ref ref="MyIdLogFile" />
 </root>
 </log4net>
 </configuration>
```

Copy the file to the Windows `SysWOW64` folder on the MyID application server.

To change the path of the log file, edit the `Log.Config` file in an text editor and update the following line:

```
<file value="c:\Logs\log.txt" />
```

Note: You must ensure that the MyID named COM user has the necessary permissions to create and write to the log file. You can create a file then give the user write permissions if you prefer not to give the user create permissions.

To disable logging, delete the `Log.Config` file from your Windows `SysWOW64` folder.

Note: Switch off logging when it is no longer needed, or you could end up with a large amount of files. The `maximumFileSize` option determines the maximum file size, but the logging will create additional files when this limit is reached.

It is important to note that this logging generates entries from all MyID components that use this form of logging.

11.3 Entrust JTK logging

You can enable logging for the Entrust JTK component. On the MyID application server, open regedit and browse to the registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\wow6432Node\Intercede\Edefice\Connector\
EntrustJTK
```

This key contains the following values:

- `JavaLocation` – an existing value containing the path to the MyID Java components.
- `LogLevel` – a DWORD value containing the logging level to use.
- `LogFile` – a String value containing the path of the JTK log file.

If the `LogLevel` or `LogFile` entries do not exist, you can create them.

For example:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Intercede\Edefice\Connector\Ent
rustJTK]
"JavaLocation"="C:\\Program Files
(x86)\\Intercede\\MyID\\Components\\Java"
"LogFile"="c:\\logs\\jtklog.log"
"LogLevel"=dword:00000004
```

In this example, the `LogFile` has been set to the logs folder on drive C:, and in a file named `jtklog.log`.

Note: Do not use the same log file as you are using for any other logging.

The logging level is set to `4`. According to the Oracle documentation for logging, the available logging levels are:

- `0` – off
- `1` – basic
- `2` – network, cache, and basic
- `3` – security, network and basic
- `4` – extension, security, network and basic
- `5` – LiveConnect, extension, security, network, temp, basic, and Deployment Rule Set

The above example will log extension, security, network, and basic calls.

To disable logging, you can set the `LogLevel` to `0`, or remove the `LogFile` entry.

For example:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Intercede\Edefice\Connector\Ent
rustJTK]
"JavaLocation"="C:\\Program Files
(x86)\\Intercede\\MyID\\Components\\Java"
"LogFile"="c:\\logs\\jtklog.log"
"LogLevel"=dword:00000000
```

or:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Intercede\Edefice\Connector\Ent
rustJTK]
"JavaLocation"="C:\\Program Files
(x86)\\Intercede\\MyID\\Components\\Java"
```

Note: The difference between providing no values and a `LogLevel` setting of `0` is that the Java tracing will create or reset the existing log file to a file of length 0, and not produce any logging.

Note: Issuing a single certificate with a `LogLevel` of `4` produces a file over 500 KB; leaving the diagnostic running has implications for disk space.

11.4 Dal4Net logging

You can configure logging on the MyID Dal4Net component. If your system uses Dal4Net – for example, for systems using SQL Azure as the database – this logs every SQL query that MyID sends to the database (but not the results of those queries).

To set up Dal4Net logging:

1. On the MyID application server, open the `Dal4Net.dll.config` file in a text editor.

   By default, this is in the following folder:

   ```
   C:\Program Files (x86)\Intercede\MyID\Components\
   Dal4Net\Dal4Net.dll.config
   ```
2. Uncomment the `<log4net>` node.
3. Update the following line to specify the location of the log file:

   ```
   <file value="Dal4Net.log"/>
   ```

Note: You must ensure that the MyID named COM user has the necessary permissions to create and write to the log file. You can create a file then give the user write permissions if you prefer not to give the user create permissions.

Important: Disable the logging when you have completed diagnosing the issues, as the log file may become very large.

12 Known Issues

- **Performance issues with antivirus scanning software**

  If you have logging switched on, MyID writes a great deal of frequently-updated data to the log file folder. With some antivirus software, this may cause a problem – under heavy load, the antivirus software checks the frequently-updated log files over and over, which may have a significant effect on the performance of your PC.

  To prevent issues occurring with your antivirus software, you are recommended to exclude the log file folder from the antivirus scanning software.