



MyID PIV
Version 12.9

Configuring Logging

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

Copyright

© 2001-2023 Intercede Limited. All rights reserved.

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

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

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

Licenses and Trademarks

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

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

Apache log4net

Copyright 2004-2021 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<https://www.apache.org/>).

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

Configuring Logging	1
Copyright	2
Conventions used in this document	3
Contents	4
1 Introduction	5
2 Windows clients	6
2.1 MyID Client WebSocket Service	8
3 MyID Image Capture	10
4 MyID Windows Integration Service	11
5 MyID Identity Agent	12
6 MyID Client Components	13
7 MyID Web Services	15
8 MyID REST and authentication web services	16
8.1 Logging Microsoft components	17
9 Other MyID web services	19
9.1 Logging for the Credential Web Service	19
9.2 Logging for the Device Management API	20
9.3 Logging for the Lifecycle API	21
10 Server components	22
10.1 Registry logging	22
10.1.1 Maximum log size and backups	25
10.1.2 Bureau logging	25
10.2 Log4Net	26
10.3 Entrust JTK logging	28
10.4 Dal4Net logging	29
11 Known issues	30

1 Introduction

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

- MyID Desktop, Self-Service App, Self-Service Kiosk and the MyID Client Service – see section [2, *Windows clients*](#).
- MyID Image Capture – see section [3, *MyID Image Capture*](#).
- MyID Windows Integration Service (WSVC) – see section [4, *MyID Windows Integration Service*](#).
- MyID Identity Agent – see section [5, *MyID Identity Agent*](#).
- MyID Client Components – see section [6, *MyID Client Components*](#).
- MyID Web Service Architecture – see section [7, *MyID Web Services*](#).
- MyID REST and Authentication Web Services – see section [8, *MyID REST and authentication web services*](#).
- Other MyID web services – see section [9, *Other MyID web services*](#).
- MyID server components – see section [10, *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.

2 Windows clients

This chapter contains information on configuring logging for the MyID Windows clients:

- MyID Desktop.
- MyID Client Service.
- Self-Service App.
- Self-Service Kiosk.
- MyID Client WebSocket Service.

Note: The MyID Client WebSocket Service uses a different method of configuring logging to the other Windows clients; see section [2.1, MyID Client WebSocket Service](#) for details.

To enable logging for these client applications, you must edit the application's configuration file.

- For MyID Desktop, the configuration file is:

`MyIDDesktop.exe.config`

and is located in the following folder by default:

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

- For the MyID Client Service, the configuration file is:

`MyIDClientService.dll.config`

and is located in the following folder by default:

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

- For the Self-Service App, the configuration file is:

`MyIDApp.exe.config`

and is located in the following folder by default:

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

- For the Self-Service Kiosk, the configuration file is:

`MyIDKiosk.exe.config`

and is located in the following folder by default:

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

To enable logging:

1. On the client PC, back up the configuration file.
2. Open the configuration file in a text editor.
3. Add the following to the `<appsettings>` section of the configuration file:

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

Set the value to true to enable logging, or false to disable logging.

By default, the log is written to the following folder:

```
%LocalAppData%\Intercede\Logs
```

If you want to specify a different location, add the following to the `<appsettings>` section of the configuration file:

```
<add key="LogDirectory" value="C:\Logs"/>
```

Set the value to the folder where you want to write the logs.

4. Save the configuration file.
5. Restart the application.

Note: If you are using the MSIX installer for your client applications, by default the logs are written to the following folder:

```
%LocalAppData%\Intercede
```

If the logs are not displayed in this folder as expected, use the `LogDirectory` setting to specify a different folder.

2.1 MyID Client WebSocket Service

You can configure logging for the MyID Client WebSocket Service by editing the service's `appsettings.json` file. By default, this file is installed in the following location:

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

```
{
  "Logging": {
    "EventLog": {
      "LogLevel": {
        "Default": "None"
      }
    }
  },
  ...
}
```

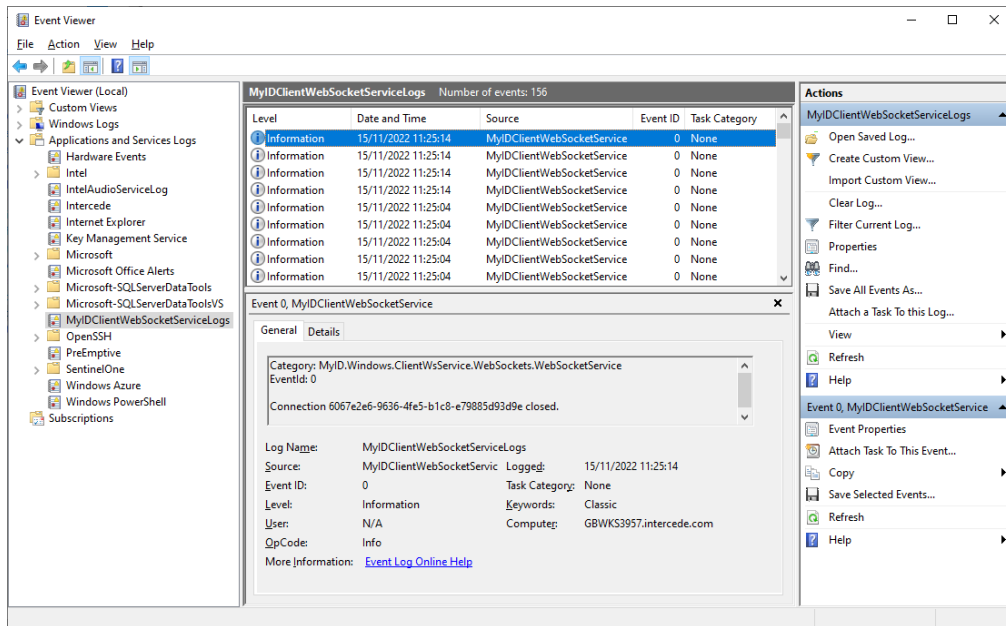
To configure logging, set the `Logging > EventLog > LogLevel > Default` value to one of the following:

- None
- Information
- Warning
- Error
- Critical
- Debug
- Trace

The options are listed in ascending level of detail.

Note: The `Trace` level causes all WebSocket messages and session ID registrations to be included in the logs, which may include sensitive information; do not use this level in a production environment.

Log information is displayed in the Windows Event Viewer, under **Applications and Services Logs > MyIDClientWebSocketServiceLogs**:



3 MyID Image Capture

To set up logging for the MyID Image Capture component, 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="MyIDApp.Utility.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.
- `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

To set up logging for the Windows Integration Service (WSVC), open the `Log.config` file, which is installed to the following folder by default:

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

Set the following options in the file:

- `configuration/log4net/appender/file` – set the value to the path and filename of the log file you want to use.

For example:

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

- `configuration/log4net/root/level` – set the value to `DEBUG`.

For example:

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

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

For example:

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

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:

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

Note: Not all of these components may be available on your system, depending on which edition of MyID you are using.

To set up logging for a component:

1. Set the following in the client PC'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 Web Services

You can set up logging for the following web services, included as part of the MyID Web Service Architecture:

- MyIDDDataSource
- MyIDProcessDriver

To set up logging:

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

For MyIDDDataSource, this is:

```
C:\Program Files\Intercede\MyID\SSP\MyIDDDataSource\Log.config
```

For MyIDProcessDriver, this is:

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

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

```
<file value="C:\logs\MyIDDDataSource.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.

8 MyID REST and authentication web services

MyID provides web services for the MyID Operator Client to communicate with and authenticate to the web server, as well as external authentication services that are used, for example, for AD FS authentication.

You can set up logging for the following web services:

- `rest.core`
- `web.oauth2`
- `web.oauth2.ext`
- `AdfsAuth`

To set up logging:

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

```
C:\Program Files\Intercede\MyID\<web service name>\Log.config
```

where `<web service name>` is the name of the web service.

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`. For diagnosing issues, you are recommended to set the level to `ERROR`; this level provides useful information without providing too much additional detail that can mask the information you need.

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.

In addition to this logging, you can also set up logging for the Microsoft components used by these web services; see section 8.1, *Logging Microsoft components* for details.

8.1 Logging Microsoft components

The logging for the REST and authentication services controlled by the `Log.config` file reports activity within MyID components; however, there may be instances within those logs where errors are reported from Microsoft components. If you require further information about these Microsoft errors, 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.Production.json` file for the web service.

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

where `<web service name>` is the name of the web service.

These files are the override configuration files for the `appsettings.json` files for the web services. If these files do not already exist, you must create them in the same folder as the `appsettings.json` files.

2. Add an entry for logging Microsoft components.

For example:

```
{
  "Logging": {
    "LogLevel": {
      "Microsoft": "Error"
    }
  }
}
```

Note: If you already have `appsettings.Production.json` files, add the `Logging:LogLevel:Microsoft` section to the existing file. The above example assumes that there are no other entries in the file.

This example adds logging information from all Microsoft components at `Error` level. You are recommended to use this level for diagnosing issues; this level provides useful information without providing too much additional detail that can mask the information you need.

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: You must make sure that the `Log.config` file is configured to produce a log (see section 8, [MyID REST and authentication web services](#)), 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.

9 Other MyID web services

You can set up logging for the following web services:

- Credential Web Service (CWS) – see section 9.1, *Logging for the Credential Web Service*.
- Device Management API (DWS) – see section 9.2, *Logging for the Device Management API*.
- Lifecycle API (MyIDEnroll) – see section 9.3, *Logging for the Lifecycle API*.

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.

9.1 Logging for the Credential Web Service

To set up logging for the Credential Web Service (CWS), copy the following into a text file called `Log.Config` in the following folder:

```
C:\Program Files\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>
```

9.2 Logging for the Device Management API

To set up logging for the Device Management API web service (DWS), copy the following into a text file called `Log.Config` in the following folder:

```
C:\Program Files\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>
```

9.3 Logging for the Lifecycle API

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

```
C:\Program Files\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>
```

10 Server components

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

- Logging using the registry method – see section [10.1, Registry logging](#).
- Logging using the Log4Net method – see section [10.2, Log4Net](#).
- Logging for Entrust JTK – see section [10.3, Entrust JTK logging](#).
- Logging for the Dal4Net component – see section [10.4, Dal4Net logging](#).

10.1 Registry logging

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

- AccessProfileImport
- ADDeletionSync
- ADDeletionSync
- AdjudicationEquifax
- AdjudicationOPM
- AMAGPACSCollector
- ASyncImport
- BOL_Authentication
- BOL_Certificates
- BOL_Core
- BOL_DeviceManagement
- BOL_DevicePolicy
- BOL_Devices
- BOL_ImportFromCard
- BOL_Jobs
- BOL_LDAP
- BOL_Notifications
- BOL_People
- CardScriptExtensions
- CBPACSCollector
- CertificateRevocationConnector
- CertificateSrv
- eActivIDSProcessor
- eBureauSrv
- ECardPrintX

- eConfiguration
- eCS
- Edefice_CS
- Edefice_DAL
- EdeficeBOL_PKI
- EdeficeSmartCard
- eDirectory
- eEMVDataProcessor
- eJobMaintenanceProcessor
- eJobServer
- eKeySrv
- eKeySrvPool
- Entrust_Admin
- EntrustJTKConnector
- ePkiConfig
- eStaleJobProcessor
- GEFPCACSConnector
- GEPACSConnector
- GPOBureauMessage
- HSMTTestUtility
- ImportProcessor
- JobBatch
- LUNAKeySrv
- MicrosoftConnector
- MicrosoftKeyStore
- MifareCom
- NCKeysrv
- OfflineRevocationConnector
- OpenPlatformSecurity
- PivDataProcessor
- PivTransport
- PreciseConnector
- ResyncByCounter
- SecugenConnector

- SymantecLH
- SymantecMPKIHelper
- SunOne
- THNGHooks
- Unicert

Note: Not all of these components may be available on your system, depending on which edition of MyID you are using.

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 [10.2, Log4Net](#).

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\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. Alternatively, you can configure a maximum file size and a backup location for your log files; see section [10.1.1, Maximum log size and backups](#).

10.1.1 Maximum log size and backups

You can configure a maximum log file size. When MyID attempts to write to the log file, if the current file size exceeds the maximum configured, MyID clears the log and starts again. Optionally, you can configure MyID to back up the old log file before clearing it.

Note: These settings are applied for all modules that use this logging method.

To configure a maximum log size and backups:

1. Within the `Trace` key in the MyID application server's registry, create a DWORD value named `LogFileSize`.
Set the value to the maximum size (in KB) of the log file.
2. Within the `Trace` key in the MyID application server's registry, create a DWORD value named `CreateBackups`.
Set the value to `1` to enable backups, and `0` to disable backups.
3. Within the `Trace` key in the MyID application server's registry, create a String value named `BackupLocation`.
Set the value to the name of the folder to which you want to copy the backup log files.

You can specify a backup location on a file server rather than on the local application server; however, you must ensure that the MyID COM user has write access to this folder. Backup log files are copied to this folder with an appended timestamp in their filenames.

10.1.2 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\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.

10.2 Log4Net

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

- EJBCA connector
- SymantecMPKI connector
- DigiCert ONE 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 `System32` folder on the MyID application server.

To change the path of the log file, edit the `Log.Config` file in a 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 `System32` 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.

10.3 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\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\Intercede\Edefice\Connector\EntrustJTK]
"JavaLocation"="C:\\Program Files\\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\Intercede\Edefice\Connector\EntrustJTK]
"JavaLocation"="C:\\Program Files\\Intercede\\MyID\\Components\\Java"
"LogFile"="c:\\logs\\jtklog.log"
"LogLevel"=dword:00000000
```

or:

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Intercede\Edefice\Connector\EntrustJTK]
"JavaLocation"="C:\\Program Files\\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.

10.4 Dal4Net logging

You can configure logging on the MyID Dal4Net component. If your system uses Dal4Net – for example, for systems using SQL Authentication – 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\Intercede\MyID\Components\Dal4Net\
```

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.

11 Known issues

This section contains any known issues that may occur when logging the MyID components.

- **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.