MyID

SecureVault

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

Document reference: IMP2071-05

December 2025

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.

Trademarks

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

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

Licenses

This software includes packages provided under a variety of licenses.

Boost

LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Boost.Serialization

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CATCH

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

boost-zlib-depend

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

gmock

Copyright 2008, Google Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. .

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

googletest

Copyright 2008, Google Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OpenSSL

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License.

However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Microsoft components

Copyright (c) Microsoft Corporation.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED *AS IS*, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Humanizer.dll

The MIT License (MIT)

Copyright (c) .NET Foundation and Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED *AS IS*, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Log4Net

Apache License Version 2.0, January 2004 http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and You must cause any modified files to carry prominent notices stating that You changed the files; and You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Swashbuckle.AspNetCore

The MIT License (MIT)

Copyright (c) 2016 Richard Morris

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Newtonsoft.Json

The MIT License (MIT)

Copyright (c) 2007 James Newton-King

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

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

SecureVault 1
Copyright 2
Conventions used in this document 14
Contents 15
1 Introduction 18
1.1 Change history 18
2 Installing MyID SecureVault 19
2.1 Prerequisites 20
2.1.1 Internet Information Services 20
2.1.2 TLS 20
2.1.3 .NET Core Hosting 20
2.2 Setting up the user accounts 21
2.2.1 Installation user 21
2.2.2 Web service user 21
2.2.3 Database user 22
2.3 Creating the databases 22
2.3.1 Database installation permissions 24
2.3.2 Using a custom database port 24
2.4 Trusting the installation scripts 25
2.5 Running the installation 27
2.5.1 Configuring the application pool 40
2.6 Automating the installation 42
2.6.1 Exporting the registry file 43
2.6.2 Populating the credentials in the registry file 44
2.6.3 Automating the population of credentials in the registry file 46
2.6.4 Running the installation in automation mode 48
2.7 Configuring the database connection manually 49
2.7.1 Encrypting the password 51
2.8 Uninstalling MyID SecureVault 52
2.8.1 Completely removing MyID SecureVault 52
3 Setting up authentication for the MyID SecureVault web service 53
3.1 Setting up OAuth2 authentication 54
3.1.1 Configuring the web.oauth2 server 54
3.1.2 Configuring the MyID SecureVault server 61
3.2 Setting up two-way TLS authentication 62
3.2.1 Generating a certificate 62
3.2.2 Configuring MyID SecureVault with the details of the certificate 63
3.2.3 Configuring IIS for two-way TLS 64
3.2.4 Trusting the client certificate on the MyID SecureVault server 66
3.2.5 Using TLS 1.3 67
3.2.6 Using a load balancer or reverse proxy 68
4 Configuring MyID SecureVault 71
4.1 Configuring MyID SecureVault for use without an HSM 72

4.1.1 Generating a self-signed certificate 72
4.1.2 Configuring the MyID SecureVault web service with the certificate 73
4.2 Configuring MyID SecureVault for use with an HSM 75
4.2.1 Installing MyID SecureVault with HSM support 76
4.2.2 Configuring MyID SecureVault to use an HSM with MyID CMS 77
4.2.3 Configuring MyID SecureVault to use an HSM as a standalone server 77
4.2.4 Configuring the MyID SecureVault web service to use an HSM 79
4.2.5 Configuring alternative key partitions 81
4.2.6 Considerations for setting the number of concurrent sessions 84
5 Using the MyID SecureVault API 85
5.1 Accessing the API documentation 86
5.2 Authenticating to the API 88
5.2.1 Authenticating to the API from the API documentation 89
5.2.2 Authenticating to the API from PowerShell 91
5.3 Calling the API 93
5.3.1 Calling the API using PowerShell and OAuth2 authentication 94
5.3.2 Calling the API using PowerShell and two-way TLS 95
5.3.3 Batch operations 96
6 Troubleshooting 98
6.1 Installation issues 99
6.2 Configuring logging 100
6.3 Auditing 101
6.4 Error codes 102
7 Changing passwords 107
7.1 Changing the MyID SecureVault web service user account password 108
Appendix A Configuring MyID CMS for MyID SecureVault 110
A.1 Prerequisites for integration between MyID CMS and MyID SecureVault 111
A.2 Setting up the MyID SecureVault external system 112
A.3 Setting up certificate policies to use MyID SecureVault 115
A.4 Importing keys into MyID SecureVault through MyID CMS 117
A.5 Recovering keys from MyID SecureVault through MyID CMS 117
A.6 SecureVault reports 118
A.6.1 SecureVault Usage report 118
A.6.2 SecureVault Biometric Usage report 119
A.7 Connection issues 120
Appendix B Configuring MyID CMS for Key Escrow 122
B.1 Overview 123
B.2 Prerequisites for key escrow 125
B.3 Configuring the MyID CMS for Key Escrow system 126
B.3.1 Setting up authentication 126
B.3.2 Setting up a connection to MyID SecureVault 126
B.3.3 Setting up your certificate policy 127
B.3.4 Setting up a key recovery credential profile 128
B.3.5 Setting up operator permissions 128
B.4 Configuring the production MyID CMS system 129

B.4.1 Setting up the certificate notification 129
B.4.2 Setting up a connection to MyID SecureVault 130
B.4.3 Setting up your certificate policies 131
B.4.4 Restricting access to key recovery workflows 132
Appendix C Configuring MyID CMS for biometric storage 133
C.1 Prerequisites for biometric storage 134
C.2 Configuring the options for biometric storage 135
C.3 Migrating biometric data to MyID SecureVault 136
C.4 Reporting and auditing biometric usage 137

1 Introduction

MyID SecureVault is a secure key archival module that allows you to:

- Store, generate, and recover private keys.
- Store and recover biometric data securely.

MyID SecureVault provides an API that allows you to integrate it with your own systems to provide key generation and data storage and recovery.

MyID SecureVault supports the optional use of an HSM for key generation and the protection of private keys and data.

You can also integrate MyID SecureVault with MyID CMS to provide key generation, storage, and recovery, and the storage and recovery of biometric data.

This document provides details of installing, configuring, and using MyID SecureVault.

Note: MyID SecureVault currently supports the generation, storage, and recovery of RSA 2048, 3072, and 4096 bit keys, and ECC P256, P384, and P521 curves. You can import, store, and recover (but not generate) RSA 1024 bit keys.

1.1 Change history

Version	Description
`IMP2071-01`	First version.
`IMP2071-02`	Released with MyID SecureVault version 2.0.0. Added more information on error code VLT10001. Added more information on logging configuration. Added support for ECC certificates. Added an appendix containing information about setting up a separate additional MyID CMS system to use for key escrow. Added information about configuring the MyID SecureVault application pool in IIS.
`IMP2071-03`	Added information on support for multiple instances of MyID SecureVault in MyID CMS 12.16 or later. Added information on setting up CORS.
`IMP2071-04`	Added information on using MyID SecureVault for the secure storage of biometric data. Added information on configuring MyID CMS to store biometric data in MyID SecureVault.
`IMP2071-05`	Added information on support for EFT biometric files in MyID CMS PIV 12.17 or later.

2 Installing MyID SecureVault

MyID SecureVault comprises the following components:

- A web service.
- A database to store keys and data.
- A database to store audit information.
- Optionally, server components for HSM support.

You can install all of these components on the same server, or install the web service (and optional HSM server components) on one server and the database on another. Additionally, if you are integrating MyID SecureVault with MyID CMS, you can install the components on your MyID servers.

This chapter contains information on:

- Supported operating systems and required software.

  See section *2.1*, *Prerequisites*.
- Setting up the installation and web service user accounts needed for MyID SecureVault.

  See section *2.2*, *Setting up the user accounts*.
- Creating the key database and the audit database.

  See section *2.3*, *Creating the databases*.
- Configuring your server to trust the signed installation scripts.

  See section *2.4*, *Trusting the installation scripts*.
- Running the installation program.

  See section *2.5*, *Running the installation*.
- Automating your installations.

  See section *2.6*, *Automating the installation*.
- Manually configuring the database connection.

  See section *2.7*, *Configuring the database connection manually*.
- Uninstalling the software.

  See section *2.8*, *Uninstalling MyID SecureVault*.

2.1 Prerequisites

MyID SecureVault supports the following server operating systems:

- Windows Server 2019

  MyID SecureVault has been tested with Windows Server 2019 Standard version 10.0.17763.

- Windows Server 2022

  MyID SecureVault has been tested with Windows Server 2022 Datacenter version 10.0.20348 (21H2).

2.1.1 Internet Information Services

You must have Internet Information Services (IIS) installed on your MyID SecureVault server. This is required to host the MyID SecureVault web service.

2.1.2 TLS

You must set up MyID SecureVault to operate over TLS.

2.1.3 .NET Core Hosting

For the MyID SecureVault server, download and install ASP.NET Core Runtime 8.0 Hosting Bundle.

To obtain the installation program, visit the Microsoft .NET download site:

dotnet.microsoft.com/en-us/download/dotnet/8.0/runtime

Important: You must install the Core Hosting Bundle *after* installing IIS. If you install the Core Hosting Bundle before IIS, you must repair the installation; run the Core Hosting Bundle installation program again after installing IIS.

2.2 Setting up the user accounts

You must configure your Windows user accounts before you start to install MyID SecureVault.

2.2.1 Installation user

You are recommended to carry out your installation using a domain user that is part of the local Administrator group. This ensures the correct setup and permissions for your installation.

If you are using Windows authentication for your database, this account is also used to create the tables in the MyID SecureVault databases. You must make sure that this account has DDL permissions for the MyID SecureVault databases.

2.2.2 Web service user

MyID SecureVault also uses a Windows user account to run the web service; you must create and configure this user account before you attempt to install MyID SecureVault.

Note: If you are installing MyID SecureVault on the same web server as MyID CMS, you can use the same account for both the MyID web service user and the MyID SecureVault web service user; they require the same permissions.

For the MyID SecureVault web service user:

- Create the account before installing MyID SecureVault.
- You are recommended to define the user under the organizational unit Service Accounts in the LDAP directory. Create the Service Accounts OU if it does not already exist.
- Set the user as a member of the domain group Domain Users and the local group Distributed COM Users on the web, application, and database servers.
- The account should not be a member of the Domain Admins or the Enterprise Admins domain groups.
- Set the password for the account so that it does not expire.
- Ensure the account is active (not disabled), unlocked, and does not expire.
- If the `manualGroupMembership` setting in IIS (available in the Configuration Editor in IIS, in the `system.applicationHost/applicationPools/applicationPoolDefaults/processModel` section) is set to `True` (the default is `False`), you must add the user to the IIS_IUSRS group on both the domain and the local machine.

After creating the MyID SecureVault web service account, on the MyID SecureVault web services server:

1. Run the Local Security Policy application.
2. Under Local Policies, select **User Rights Assignment**.
3. Double-click **Log on as a service**.
4. Add the MyID SecureVault web service user, then click **OK** to save the changes.

Note: The web service user account requires the **Log on as a batch job** privilege – make sure that the group policy does not remove the privilege.

2.2.3 Database user

For the database user, you can use Windows Authentication using the MyID SecureVault web service user, or use SQL Authentication.

See section *2.3*, *Creating the databases* for details.

2.3 Creating the databases

MyID SecureVault uses SQL Server databases to store its keys and data and to record its audit information. You can use the same database for both purposes, or you can use separate databases.

MyID SecureVault supports the following database platforms:

- SQL Server.
- Microsoft Azure SQL.
- Amazon RDS for SQL Server.

MyID SecureVault supports the following SQL Server versions:

- SQL Server 2022 – RTM-CU20 (16.0.4205.1 July 2025)
- SQL Server 2019 – RTM-CU32 (15.0.4430.1 February 2025)

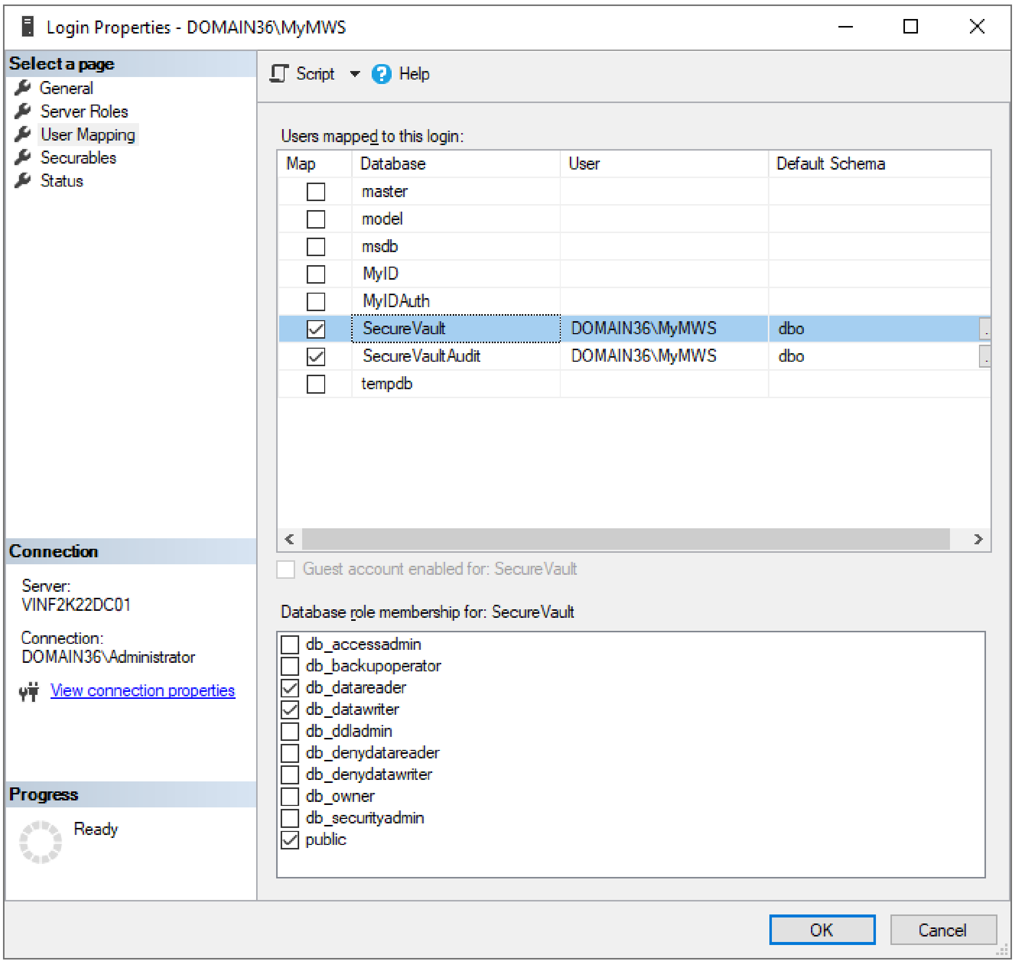
Note: Intercede supports the database versions listed above. If you want to use different service packs or cumulative updates for these major versions than those listed above, make sure that you carry out additional testing within your environment. For production deployment, SQL Server Enterprise or Standard editions must be used. Do not use major versions that are not listed above (for example, SQL Server 2012) as these are not supported.

You must create the databases you want to use before you run the MyID SecureVault installation program; the installation program does not create any databases. You can use any names for the databases, but you are recommended to use the following:

- `SecureVault`
- `SecureVaultAudit`

You must configure a login for each database that has the following permissions:

- `db_datareader`
- `db_datawriter`
- `public`



Server:
VINF2K22DC01

Connection:
DOMAIN36\Administrator

Users mapped to this login:

Map	Database	User	Default Schema
☐	master		
☐	model		
☐	msdb		
☐	MyID		
☐	MyIDAuth		
☑	SecureVault	DOMAIN36\MyMWS	dbo
☑	SecureVaultAudit	DOMAIN36\MyMWS	dbo
☐	tempdb		

☐ Guest account enabled for: SecureVault

Database role membership for: SecureVault

☐ db_accessadmin
☐ db_backupoperator
☑ db_datareader
☑ db_datawriter
☐ db_ddladmin
☐ db_denydatareader
☐ db_denydatawriter
☐ db_owner
☐ db_securityadmin
☑ public

You can use Windows Authentication using the MyID SecureVault web service user (see section 2.2, *Setting up the user accounts* for details of creating the user account) or SQL Authentication for the login; note, however, that if you are using Microsoft Azure SQL or Amazon RDS for SQL Server as your database, you *must* use SQL Authentication; you cannot use Windows Authentication.

2.3.1 Database installation permissions

The user account that runs the SQL scripts to create the tables in the MyID SecureVault databases must have Data Definition Language (DDL) permissions.

If you are using Windows Authentication for your databases, the user account that you use to run the installation process creates the tables in the MyID SecureVault databases; make sure that this account has DDL permissions on the databases.

If you are using SQL Authentication for your databases, the SQL user accounts that you specify during the installation process create the tables in the MyID SecureVault databases; make sure that these accounts have DDL permissions on the databases.

If you are using SQL Authentication, but prefer not to grant these users DDL permissions, you can use the following process:

1. Run the installation program.

   See section *2.5*, *Running the installation* for details.
2. Do *not* select the **Database** component.
3. On the database screens, provide the SQL Authentication details of the user accounts you want to use for the databases.
4. Complete the installation.

   The installer sets up the connection strings for the databases using the SQL Authentication details you provided, but does *not* create the database tables.
5. As a DBA, manually run the SQL scripts against your databases:
   a. Make sure you are logged in as a database administrator with DDL permissions.
   b. Navigate to the following folder:

      ```
      <install folder>\Installer\SECUREVAULT-<version>\
      Scripts\ConfigurationDBScripts\Extra
      ```
   c. Run the following script against the MyID SecureVault database:

      ```
      SecureVaultDB.sql
      ```
   d. Run the following script against the MyID SecureVault audit database:

      ```
      AuditDB.sql
      ```

2.3.2 Using a custom database port

If your database server uses a port other than the default TCP 1433, you must configure this manually in the registry before you install MyID SecureVault.

To set the port:

1. In the Registry Editor, navigate to:

   ```
   HKLM\SOFTWARE\Intercede\Edefice\SecureVaultInstallation\Properties
   ```
2. Create a String Value with the name:

   ```
   SQL_Port
   ```
3. Set the value to the port you want to use.

2.4 Trusting the installation scripts

The scripts that are used in the installation process are signed to confirm that they were provided by Intercede and have not been altered. If your system is configured to allow only signed PowerShell scripts to be run, you must trust Intercede as a publisher before you run the installation program. If you do not follow these instructions, the scripts will not run.

To trust Intercede as a publisher:

1. In the following folder:

   ```
   <install folder>\Support Tools\SecureVaultInstallationAssistant
   ```

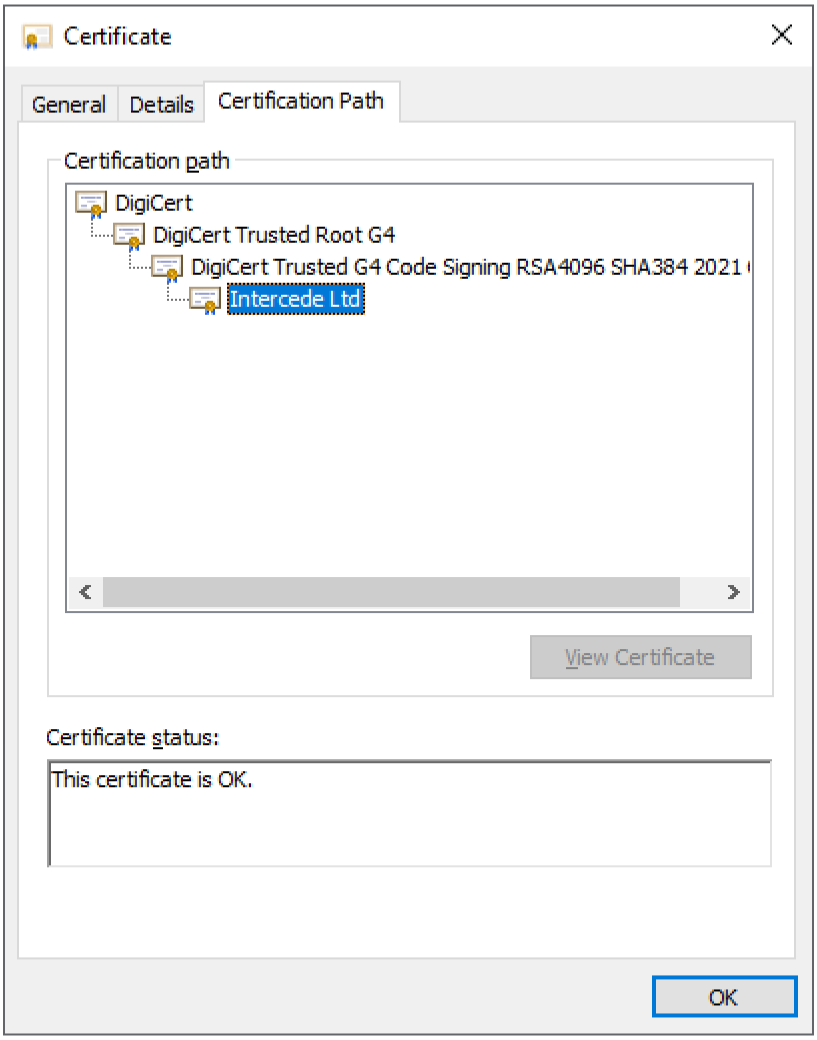
   right-click the following script:

   ```
   SecureVaultInstallationAssistant.ps1
   ```

   and from the pop-up menu, select **Properties**.
2. In the Properties dialog, click the **Digital Signatures** tab.
3. Select the Intercede signature in the signing list, and click **Details**.
4. Click **View Certificate**.
5. Click **Install Certificate**.
6. Select **Local Machine**, and click **Next**.
7. Select **Place all certificates in the following store**, click **Browse**, select the **Trusted Publishers** store, and click **OK**.
8. Click **Next**, then click **Finish**.

   The certificate is now installed to the Trusted Publishers store.

Note: The signing certificate depends on DigiCert certificates.



DigiCert
DigiCert Trusted Root G4
DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021
Intercede Ltd

This certificate is OK.

You must obtain the DigiCert certificates listed in the **Certification path** from DigiCert if your server does not already have them.

2.5 Running the installation

Note: Currently you cannot upgrade an installation of MyID SecureVault. You must uninstall the previous version before installing the new version.

To install MyID SecureVault:

1. Extract your MyID SecureVault installation files to a new folder.

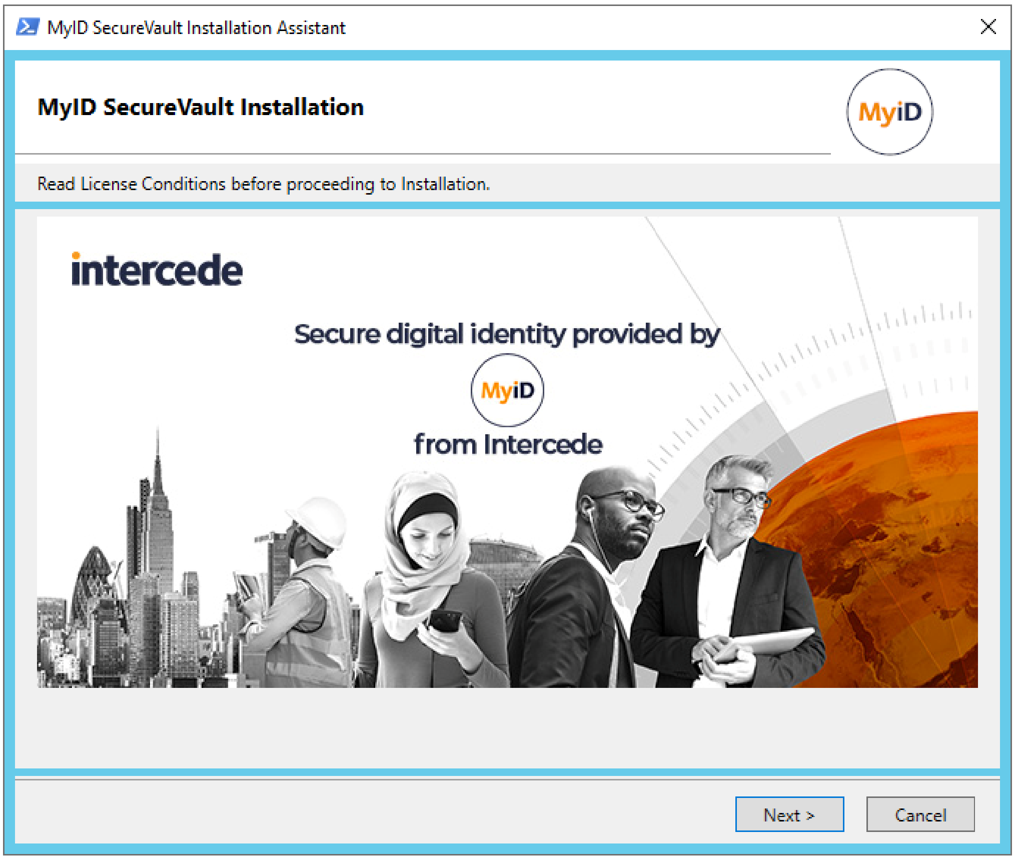
   Important: The MyID SecureVault uninstallation process requires PowerShell scripts that are provided as part of the MyID SecureVault installation package. If you move or delete the installation folder, you will be unable to uninstall MyID SecureVault using the Windows Control Panel **Programs and Features** option; when it is unable to locate the scripts, the uninstallation process displays an error. You are strongly recommended to retain the MyID SecureVault installation folder in the location from which you originally ran the installation program; this may influence your choice of folder from which to install MyID SecureVault .

   You must also continue to use the same installation folder for any future updates or upgrades.
2. Open a Windows PowerShell prompt as an administrator, and navigate to the folder where you extracted the installation files.
3. Navigate to the following folder:

   ```
   <install folder>\Support Tools\SecureVaultInstallationAssistant\
   ```
4. Run the following PowerShell script:

   ```
   .\SecureVaultInstallationAssistant.ps1
   ```

   The MyID SecureVault installation program opens.



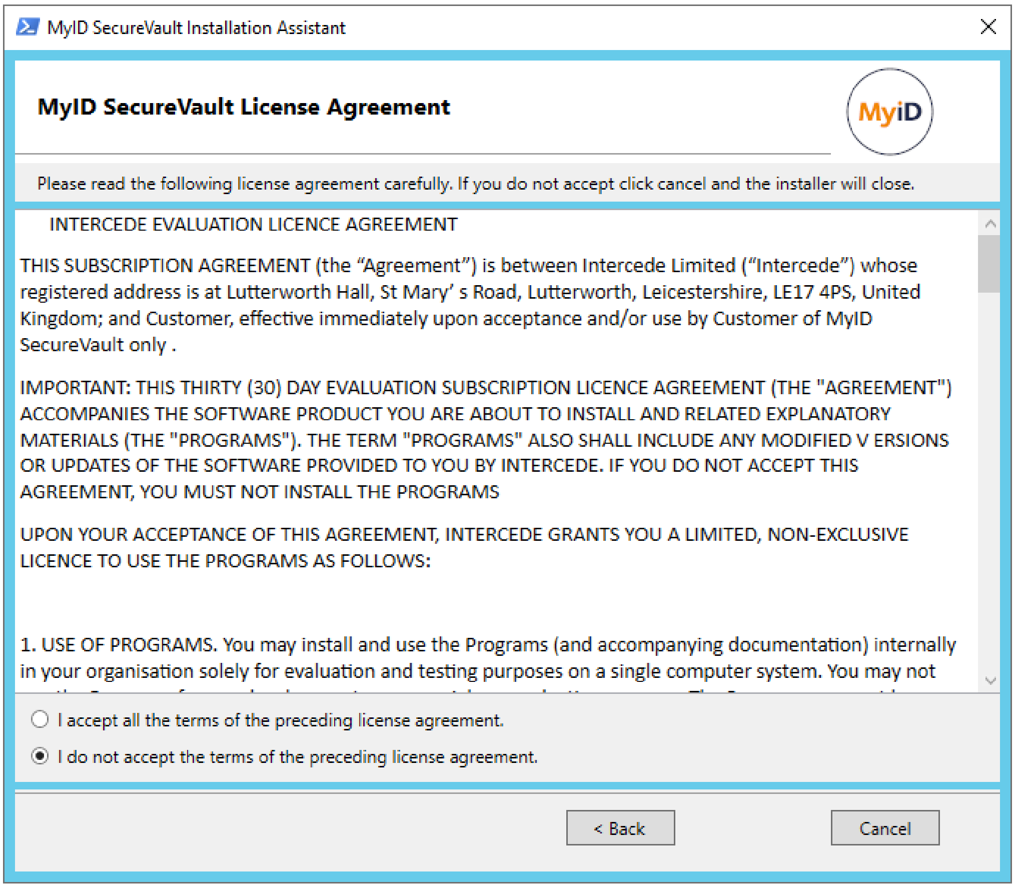
MyID SecureVault Installation Assistant

MyID SecureVault Installation

Read License Conditions before proceeding to Installation.

5. Click **Next**.

   The license agreement screen appears.



MyID SecureVault License Agreement

Please read the following license agreement carefully. If you do not accept click cancel and the installer will close.

INTERCEDE EVALUATION LICENCE AGREEMENT

THIS SUBSCRIPTION AGREEMENT (the “Agreement”) is between Intercede Limited (“Intercede”) whose registered address is at Lutterworth Hall, St Mary’ s Road, Lutterworth, Leicestershire, LE17 4PS, United Kingdom; and Customer, effective immediately upon acceptance and/or use by Customer of MyID SecureVault only .

IMPORTANT: THIS THIRTY (30) DAY EVALUATION SUBSCRIPTION LICENCE AGREEMENT (THE "AGREEMENT") ACCOMPANIES THE SOFTWARE PRODUCT YOU ARE ABOUT TO INSTALL AND RELATED EXPLANATORY MATERIALS (THE "PROGRAMS"). THE TERM "PROGRAMS" ALSO SHALL INCLUDE ANY MODIFIED V ERSIONS OR UPDATES OF THE SOFTWARE PROVIDED TO YOU BY INTERCEDE. IF YOU DO NOT ACCEPT THIS AGREEMENT, YOU MUST NOT INSTALL THE PROGRAMS

UPON YOUR ACCEPTANCE OF THIS AGREEMENT, INTERCEDE GRANTS YOU A LIMITED, NON-EXCLUSIVE LICENCE TO USE THE PROGRAMS AS FOLLOWS:

1. USE OF PROGRAMS. You may install and use the Programs (and accompanying documentation) internally in your organisation solely for evaluation and testing purposes on a single computer system. You may not

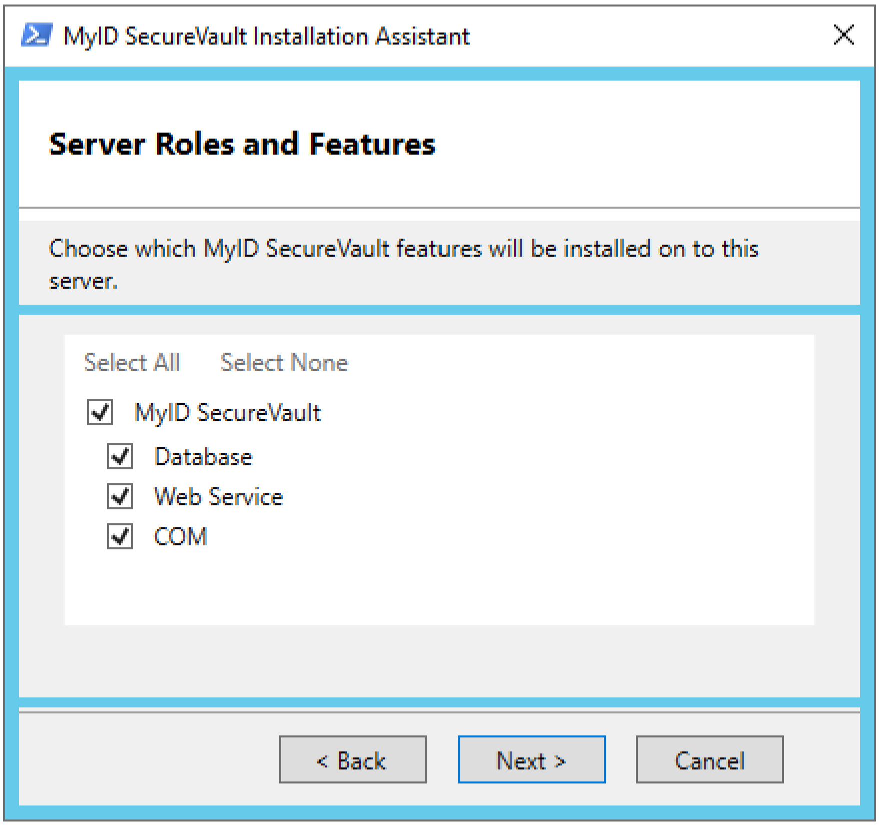
○ I accept all the terms of the preceding license agreement.

● I do not accept the terms of the preceding license agreement.

< Back   Cancel

6. Click **I accept all the terms of the preceding license agreement** and click **Next**.

   The Server Roles and Features screen appears.



MyID SecureVault Installation Assistant

Server Roles and Features

Choose which MyID SecureVault features will be installed on to this server.

☑ MyID SecureVault

☑ Database

☑ Web Service

☑ COM

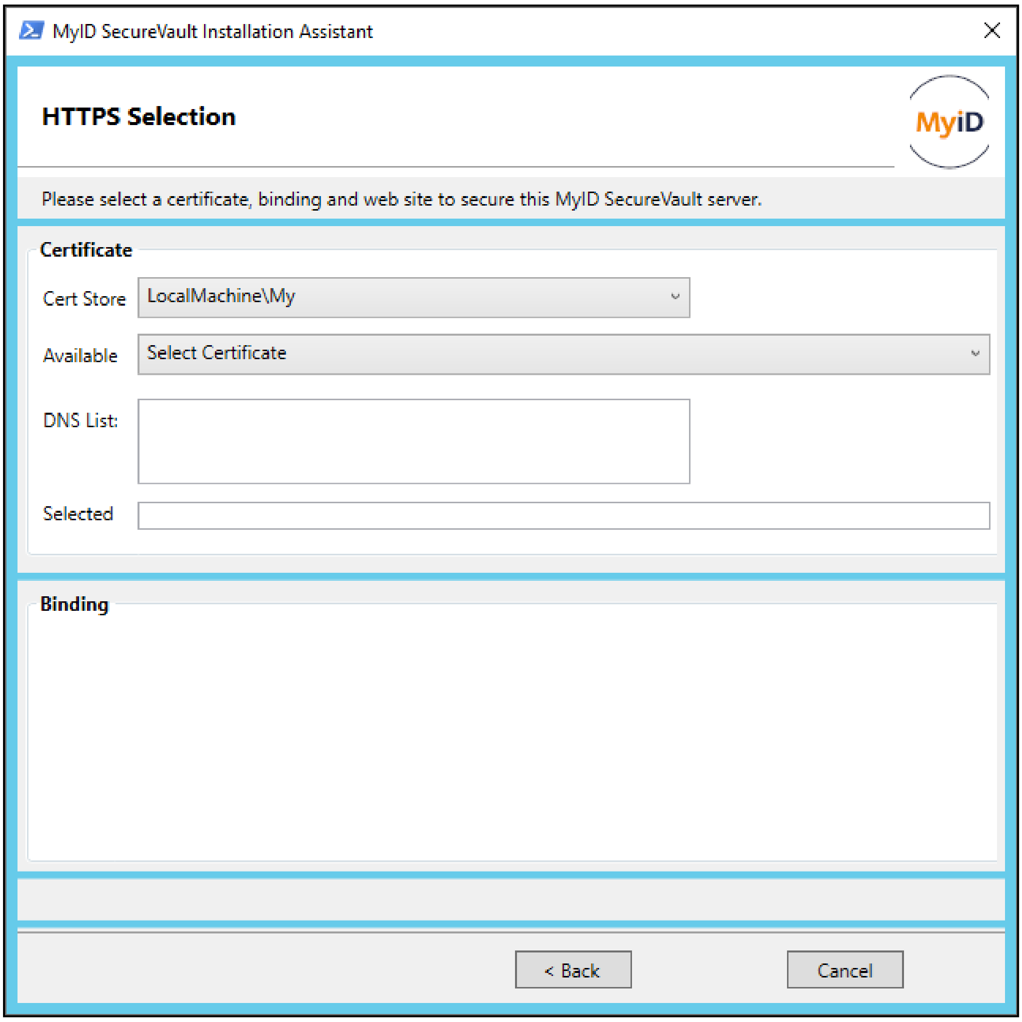
7. Select the options you want to install:
   - **Database** – installs both the MyID SecureVault key and data database and the MyID SecureVault audit database.

     You can install the databases on the same server as the web service, or on a different server. If you want to install the database on a different server, you are recommended to install the databases on the remote server *before* you install the web services.

     You can also install the databases on a different server at the same time as you install the web service; when you choose the location of the database, select a remote server instead of the local server.
   - **Web Service** – installs the MyID SecureVault web service.
   - **COM** – installs the optional application components that allow HSM access. Install these components on the same server as the web service.

     Important: If you are installing MyID SecureVault onto the same server as the MyID CMS application server, *do not* install the COM components; the MyID CMS application server already contains these components, and MyID SecureVault can make use of them. If you install the COM components on a MyID application server, and then subsequently uninstall MyID SecureVault, you may damage your MyID CMS installation.
8. Click **Next**.

   If you are installing the web service, the HTTPS Selection screen appears.



Please select a certificate, binding and web site to secure this MyID SecureVault server.

Certificate

Cert Store: LocalMachine\My

Available: Select Certificate

DNS List:

Selected:

Binding

The MyID SecureVault web service requires https, which means that you must select a certificate and bind it to the IIS website that contains the web service. You must request, install, and bind the certificate before you start to install MyID SecureVault.

Note: If you are installing MyID SecureVault on the same web server as MyID CMS, you can use the same certificate and binding that you are using for the MyID website.

For a pre-production system, you can use a self-signed certificate. To add a self-signed certificate and bind it to the website:

a. In Internet Information Services (IIS) Manager, select the server from the tree.

b. Double-click **Server Certificates**.

c. In the **Actions** pane, click **Create Self-Signed Certificate**.

d. Type a name for the certificate, then click **OK** to add it to the Personal store.

e. Select the **Default Web Site** in the tree.

f. In the **Actions** pane, select **Bindings**.

g. Click **Add**.

h. From the **Type** drop-down list, select `https`.

i. In the **Port** box, type the port number you want to use.

   The default for https is `443`. You cannot use a port that is already in use.

j. From the **SSL certificate** drop-down list, select the self-signed certificate you created.

k. Click **OK**.

To select a certificate and binding:

a. From the **Cert Store** drop-down list, select the certificate store that contains the certificate you want to use for https.

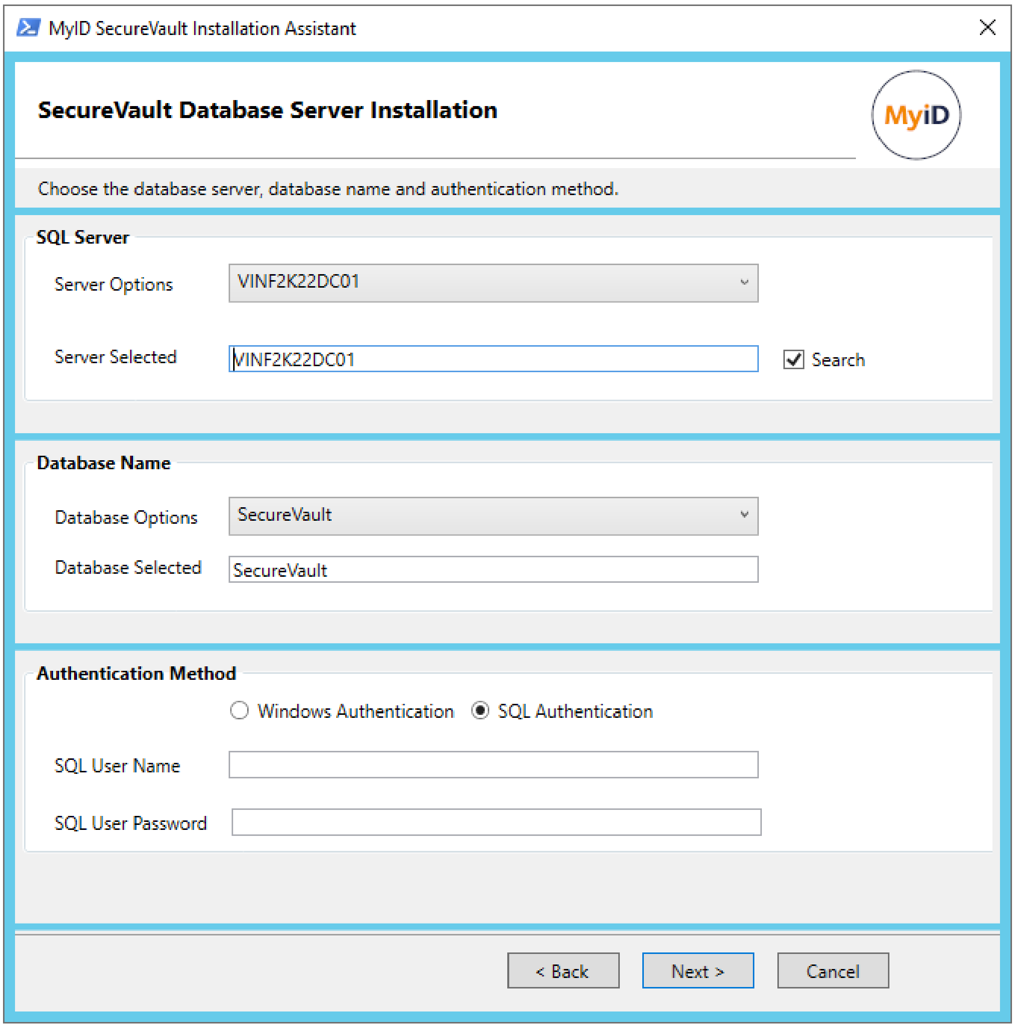
b. From the **Available** drop-down list, select the certificate you want to use for https.

c. In the **Binding** section, from the **Available** drop-down list, select the website binding you want to use for the MyID SecureVault web service.

   Note: If the website you selected also supports http, the installation program displays a warning. You are recommended to remove the http binding on the website and use https only.

d. Click **Next**.

The SecureVault Database Server Installation screen appears.



MyID SecureVault Installation Assistant

SecureVault Database Server Installation

Choose the database server, database name and authentication method.

SQL Server

Server Options: VINF2K22DC01

Server Selected: VINF2K22DC01 ☑ Search

Database Name

Database Options: SecureVault

Database Selected: SecureVault

Authentication Method

○ Windows Authentication ● SQL Authentication

SQL User Name:

SQL User Password:

Important: This screen appears whether or not you are installing the database components; you must provide the connection details for the database to the web service. If you are installing the web service but have not yet installed the database, you can click **Skip** to skip this stage, but you must configure the connection manually once you have created the database; see section *2.7*, *Configuring the database connection manually*. For this reason, you are recommended to create the database before you install the web service, or at the same time as you install the web service.

9. In the **SQL Server** section, select the database server from the **Server Options** drop-down list.

   This list provides the results of a search for data sources.

   Alternatively, you can deselect the **Search** option and type the server name manually.

   If your database server uses a named instance, type the name as `server\instance` – for example:

   `MYSERVER\myinstance`

   Note: If your database server uses a port other than the default TCP 1433, you must configure this manually in the registry; see section *2.3.2*, *Using a custom database port*.

10. In the **Database Name** section, select the database you want to use the store the MyID SecureVault keys and data from the **Database Options** drop-down list.

    Note: The database must already exist. See section *2.3*, *Creating the databases*.

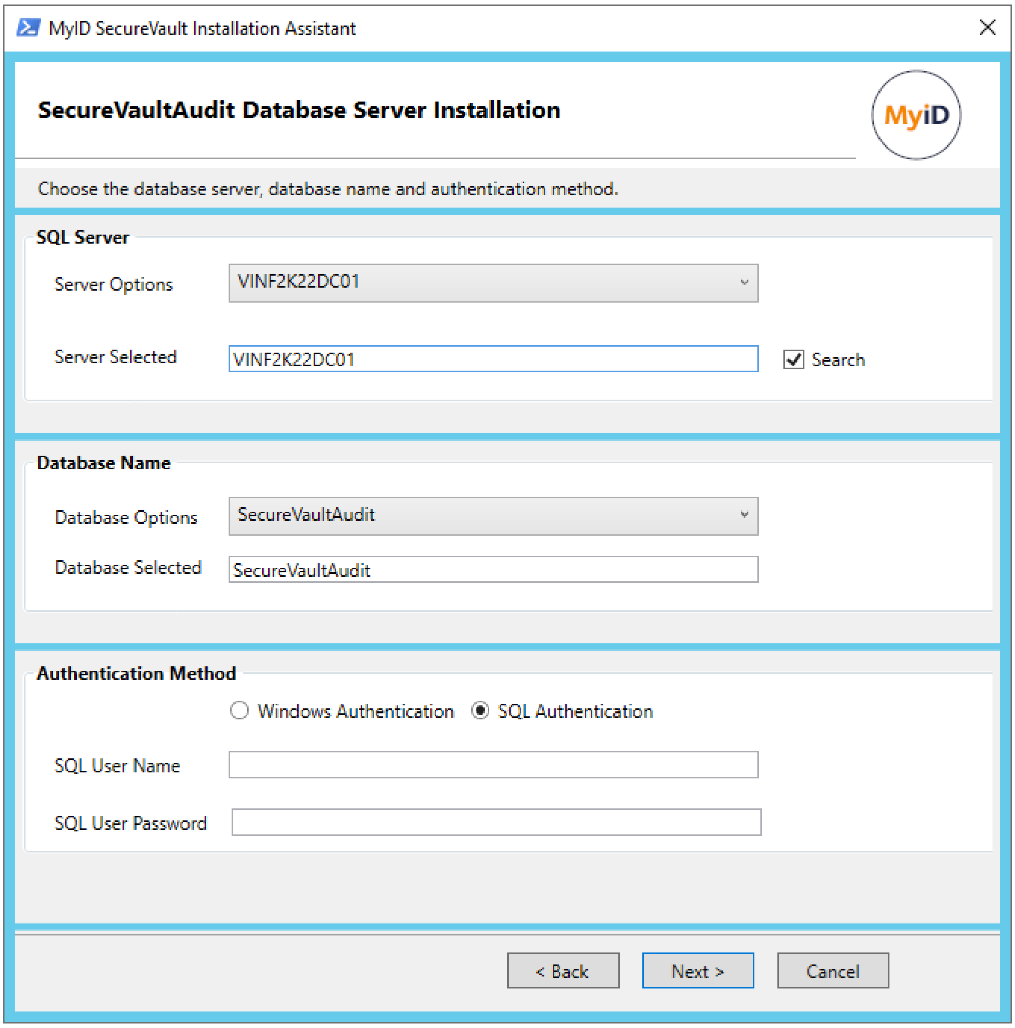
11. In the **Authentication Method** section, select one of the following:

    - **Windows Authentication** – the user account used to run the MyID SecureVault web service is used to access the SQL Server database. If the database is on a separate server on the domain to the web service, this must be a domain account.
    - **SQL Authentication** – you must specify the **SQL User Name** and **SQL User Password** for the user you want to use to authenticate to the SQL Server database.

      Note: For SQL Authentication, you must create the login account before running the installation program. See section *2.2*, *Setting up the user accounts* for details.

12. Click **Next**.

    The SecureVaultAudit Database Server Installation screen appears.



MyID SecureVault Installation Assistant

SecureVaultAudit Database Server Installation

Choose the database server, database name and authentication method.

SQL Server

Server Options: VINF2K22DC01

Server Selected: VINF2K22DC01 ☑ Search

Database Name

Database Options: SecureVaultAudit

Database Selected: SecureVaultAudit

Authentication Method

○ Windows Authentication ● SQL Authentication

SQL User Name:

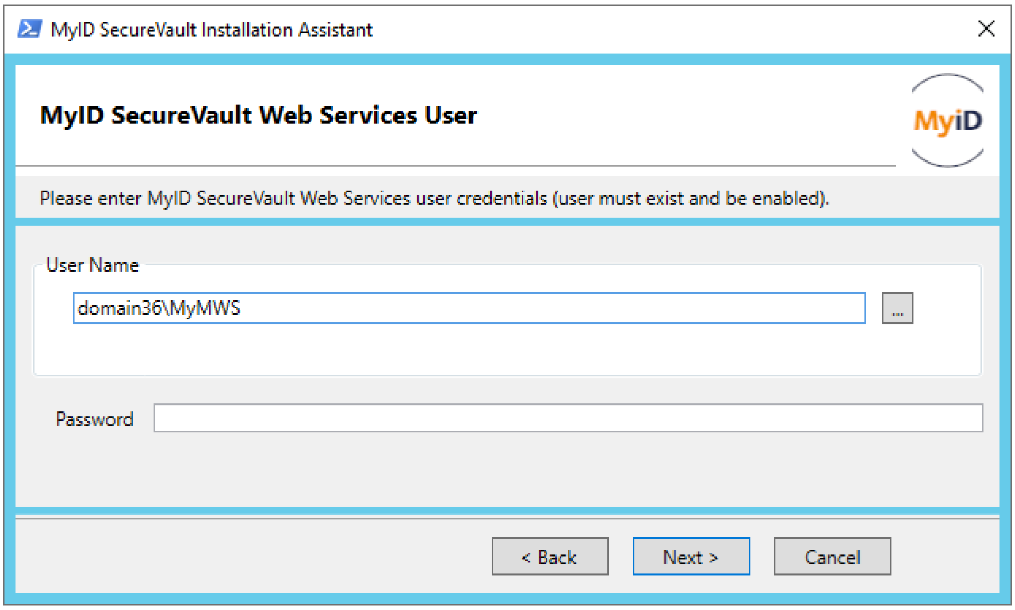
SQL User Password:

13. Provide the details for your MyID SecureVault audit database.

You must provide the same information as you have already provided for the main keys and data database, except you must specify the database you created to store the audit information.

14. Click **Next**.

    The MyID SecureVault Web Services User screen appears.



MyID SecureVault Installation Assistant

MyID SecureVault Web Services User

Please enter MyID SecureVault Web Services user credentials (user must exist and be enabled).

User Name: domain36\MyMWS

Password:

15. Provide the Windows user you want to use to run the MyID SecureVault web service.

    You can either type the domain and username; for example:

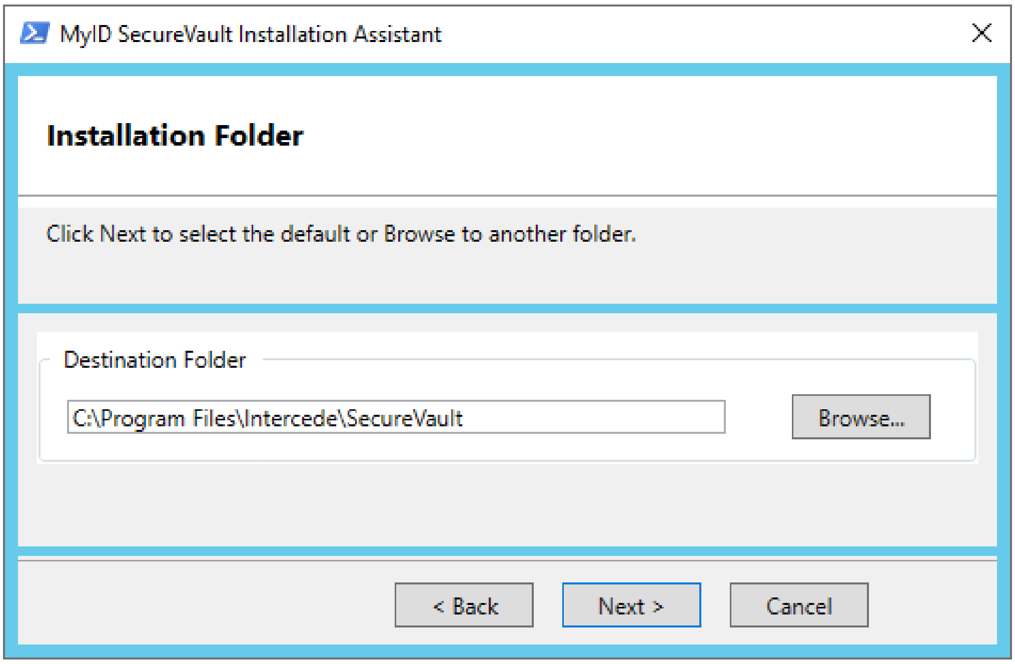
    ```
    MYDOMAIN\MyIDSecureVaultUser
    ```

    or click the **...** button to browse your Active Directory.

    See section *2.2*, *Setting up the user accounts* for details.

16. Type the user's **Password** and click **Next**.

    The Installation Folder screen appears.



MyID SecureVault Installation Assistant

Installation Folder

Click Next to select the default or Browse to another folder.

Destination Folder

C:\Program Files\Intercede\SecureVault

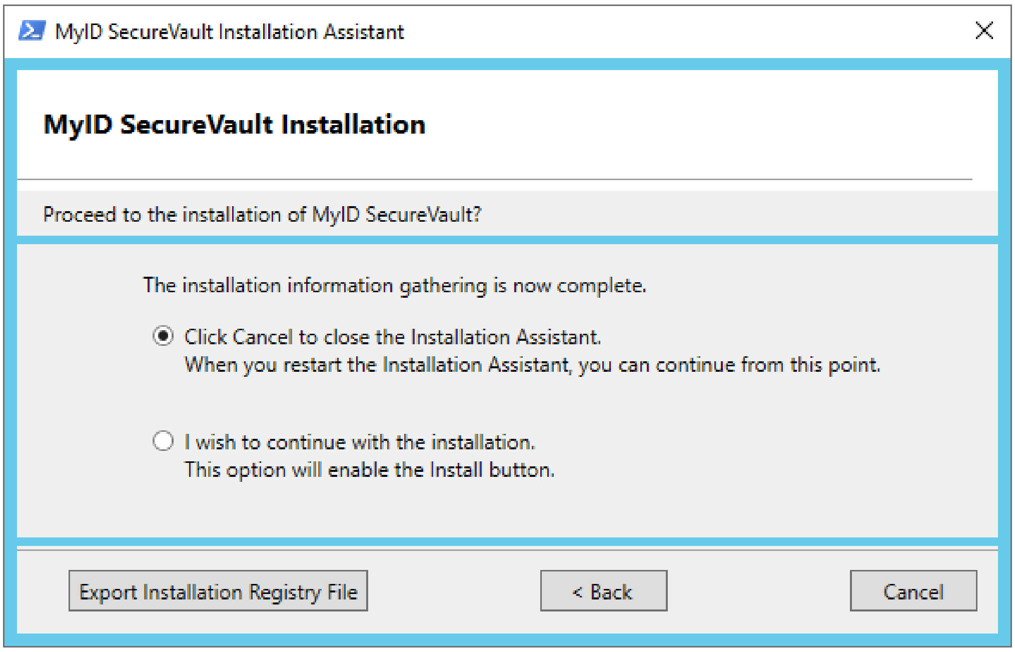
17. Select the folder into which you want to install the MyID SecureVault components.

    By default, the installation folder is:

    `C:\Program Files\Intercede\SecureVault`

18. Click **Next**.

    The installation options screen appears.



MyID SecureVault Installation Assistant

MyID SecureVault Installation

Proceed to the installation of MyID SecureVault?

The installation information gathering is now complete.

● Click Cancel to close the Installation Assistant.
When you restart the Installation Assistant, you can continue from this point.

○ I wish to continue with the installation.
This option will enable the Install button.

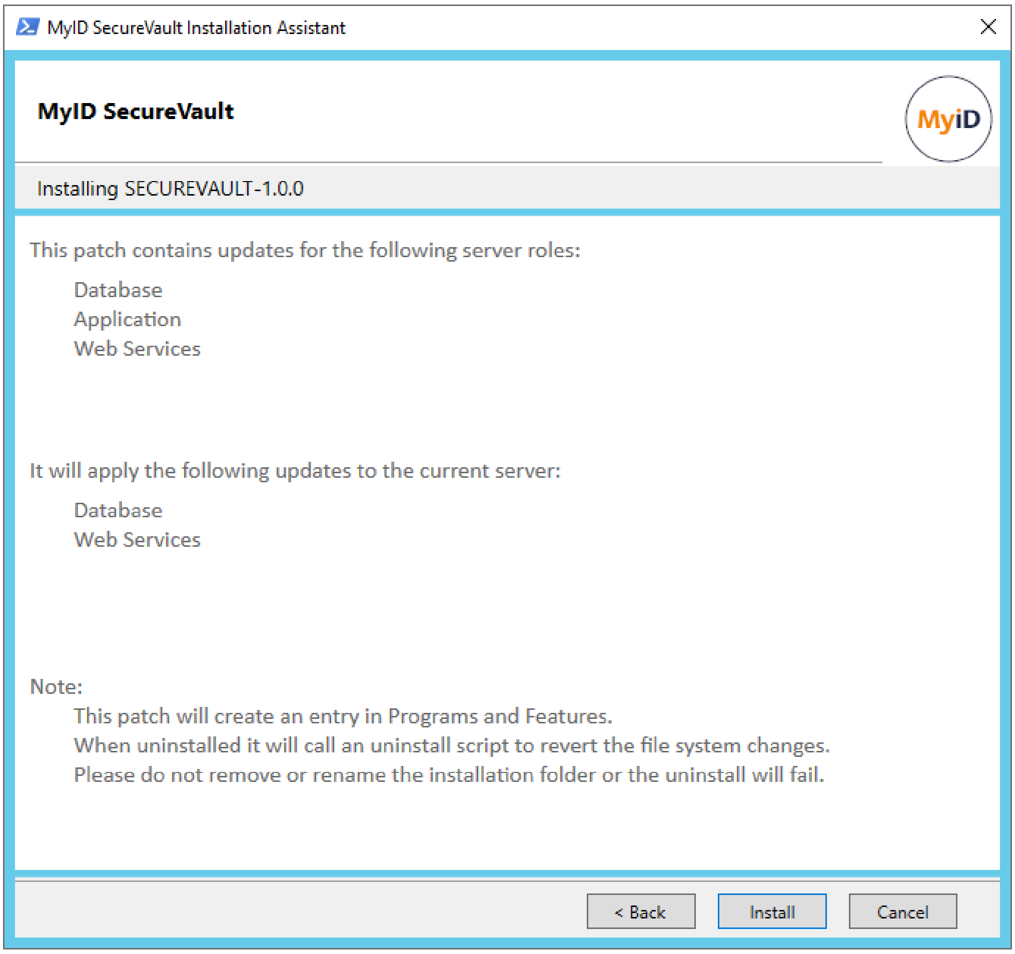
19. Carry out one of the following actions:
    - Click **Cancel**.

      The installation program closes, but when you run it again, you can continue from this point.
    - Click **Export Installation Registry File**.

      The installation program exports a registry file that you can use to automate installations of MyID SecureVault on this server or on other servers. See section *2.6, Automating the installation*.
    - Click **I wish to continue with the installation**.

      The **Install** button appears. Click **Install** to proceed with the installation.

    When you click **Install**, the installation program displays the installation confirmation screen.


MyID SecureVault Installation Assistant
MyID SecureVault
MyID
Installing SECUREVAULT-1.0.0
< Back
Install
Cancel

This patch contains updates for the following server roles:

Database
Application
Web Services

It will apply the following updates to the current server:

Database
Web Services

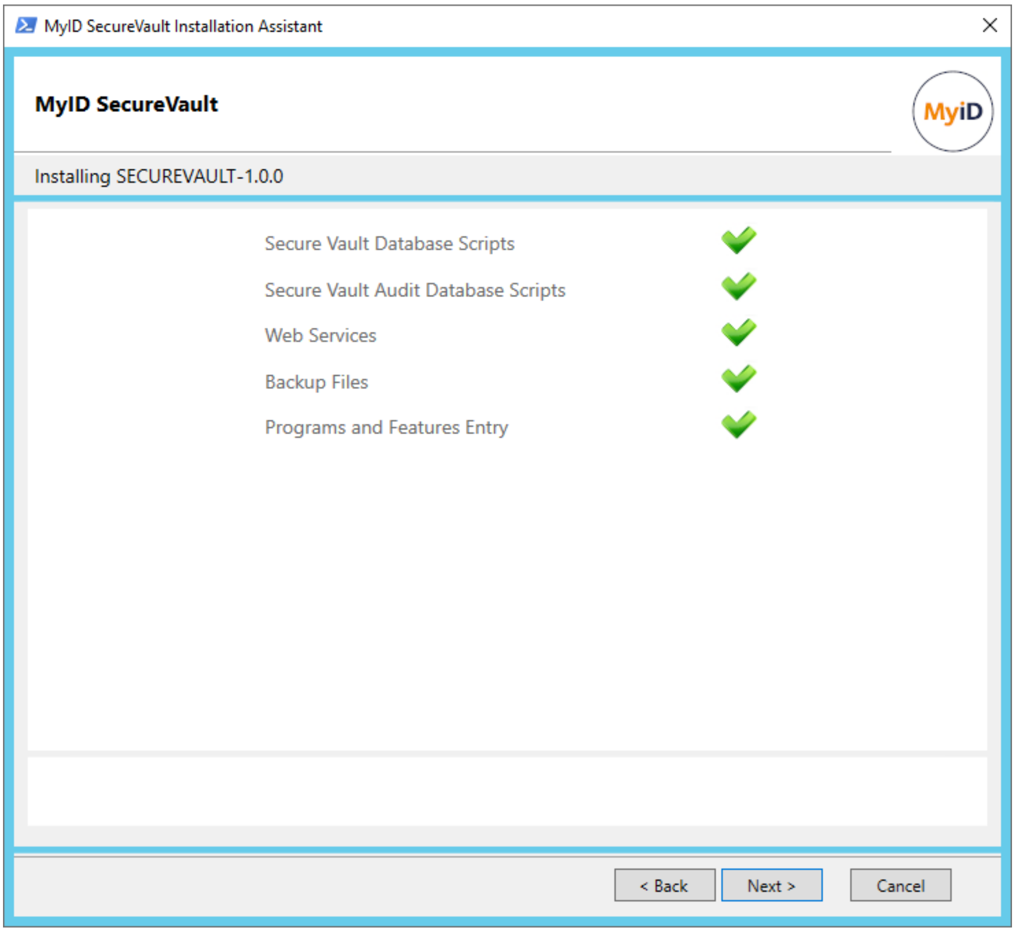
Note:

This patch will create an entry in Programs and Features.
When uninstalled it will call an uninstall script to revert the file system changes.
Please do not remove or rename the installation folder or the uninstall will fail.

This screen confirms which components you have selected for installation.

20. Click **Install** to begin the installation.

    The installation progress screen appears.



MyID SecureVault Installation Assistant

MyID SecureVault

Installing SECUREVAULT-1.0.0

Secure Vault Database Scripts ✓

Secure Vault Audit Database Scripts ✓

Web Services ✓

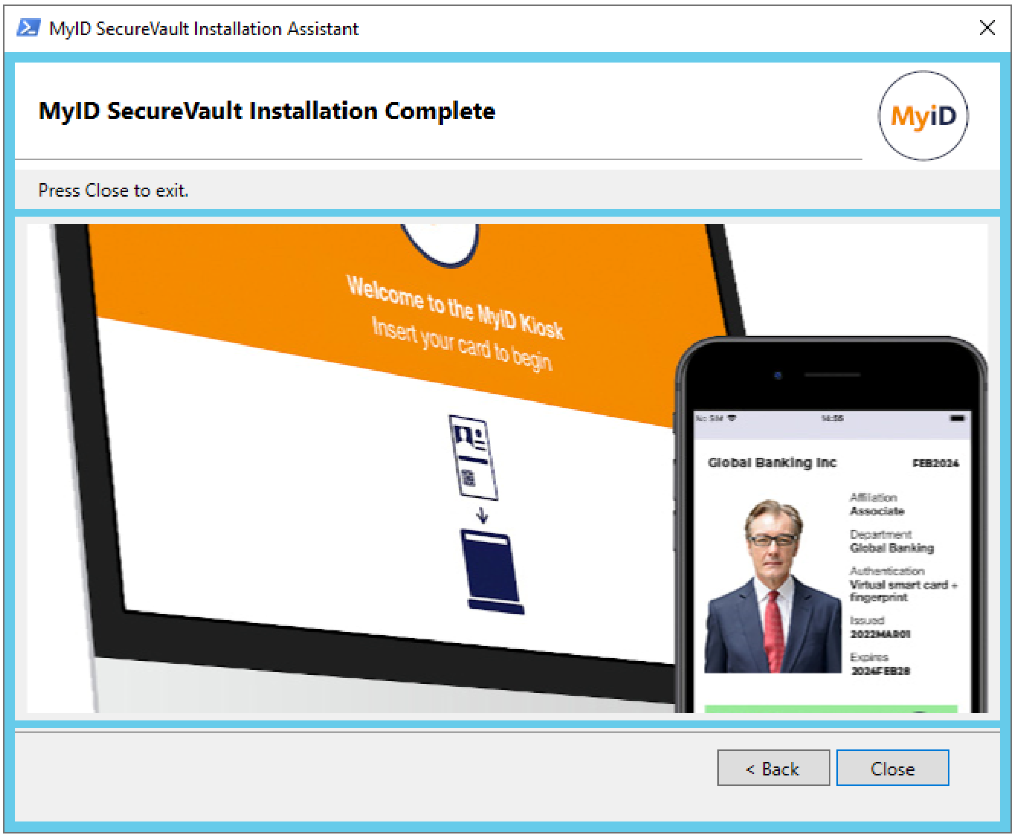
Backup Files ✓

Programs and Features Entry ✓

If an issue occurs during the installation, the installation stops at the stage where the issue occurred, and the installer displays an error indicator. See section *6.1*, *Installation issues* for assistance with errors that may occur during installation.

21. When the installation has completed, click **Next**.

    The installation complete screen appears.



MyID SecureVault Installation Assistant

MyID SecureVault Installation Complete

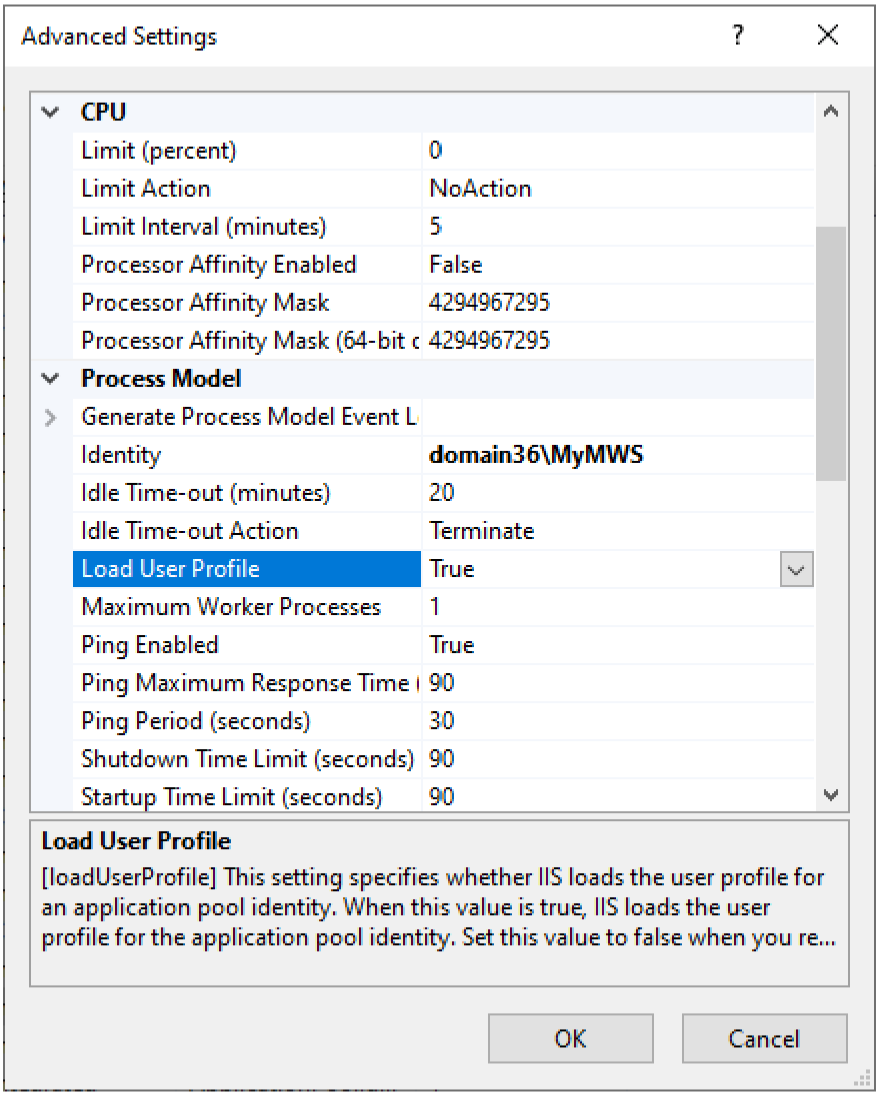
Press Close to exit.

22. Click **Close**.

2.5.1 Configuring the application pool

You must set the **Load User Profile** option for the MyID SecureVault application pool to **True**.

1. On the MyID SecureVault web server, in Internet Information Services (IIS) Manager, select **Application Pools**.
2. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Advanced Settings**.
3. In the **Process Model** section, set the **Load User Profile** option to **True**.



Advanced Settings

CPU	
Limit (percent)	0
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	False
Processor Affinity Mask	4294967295
Processor Affinity Mask (64-bit c	4294967295
Process Model	
Generate Process Model Event L	
Identity	**domain36\MyMWS**
Idle Time-out (minutes)	20
Idle Time-out Action	Terminate
Load User Profile	True
Maximum Worker Processes	1
Ping Enabled	True
Ping Maximum Response Time (	90
Ping Period (seconds)	30
Shutdown Time Limit (seconds)	90
Startup Time Limit (seconds)	90

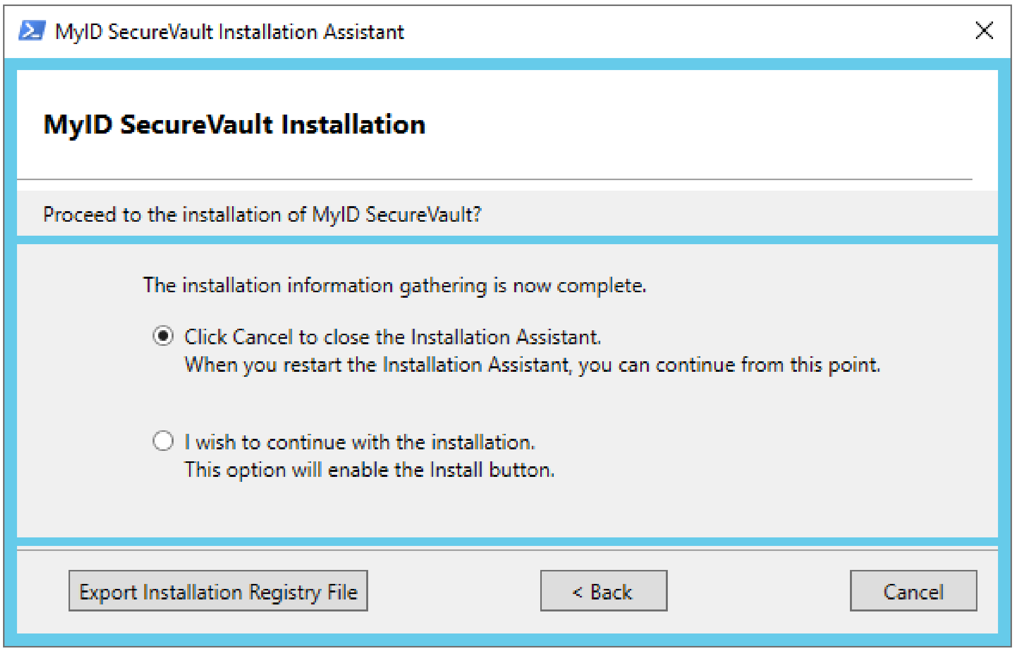
Load User Profile
[loadUserProfile] This setting specifies whether IIS loads the user profile for an application pool identity. When this value is true, IIS loads the user profile for the application pool identity. Set this value to false when you re...

4. Click **OK**.

Note: If you do not set this option, you may experience an error when generating a key in the ServerKeyStore similar to the following:

```
Error Generating key in ServerKeyStore - Info: POST to:
https://react.domain37.local/securevault/api/Keys/68ef5dd3-b3ba-4c4a-a6e5-
6e031b0496e5/sign failed, response:
{"type":"https://forums.intercede.com/errorCodes/SecureVault#VLT10001","ti
tle":"A protection error has occurred.","status":500,"code":"VLT10001"} ---
---------------------- Exception raised in function:
SendHttpRequest::SendHTTPRequest In file SendHttpRequest.cpp at line 116
(std)
```

2.6 Automating the installation



MyID SecureVault Installation Assistant

MyID SecureVault Installation

Proceed to the installation of MyID SecureVault?

The installation information gathering is now complete.

● Click Cancel to close the Installation Assistant.
When you restart the Installation Assistant, you can continue from this point.

○ I wish to continue with the installation.
This option will enable the Install button.

Once you have reached the installation options screen, you can decide to proceed with the installation, or close the installer to return to it at a later time and complete the installation; alternatively, you can export all the settings you have entered up to this point as a registry file that you can use to automate installations of MyID SecureVault on this server or on other servers.

The process is as follows:

1. Export the registry file.

   See section *2.6.1*, *Exporting the registry file*.

2. Update the registry file with the credentials.

   The registry file does not contain any of the usernames or passwords you have provided when going through the installation process.

   You must populate the file with the appropriate credentials before you can use it to automate an installation. You can:

   - Provide the credentials interactively.

     See section *2.6.2*, *Populating the credentials in the registry file*.

   - Create a configuration file to store the credentials.

     See section *2.6.3*, *Automating the population of credentials in the registry file*.

3. Run the MyID SecureVault installation program with the automation parameter.

   See section *2.6.4*, *Running the installation in automation mode*.

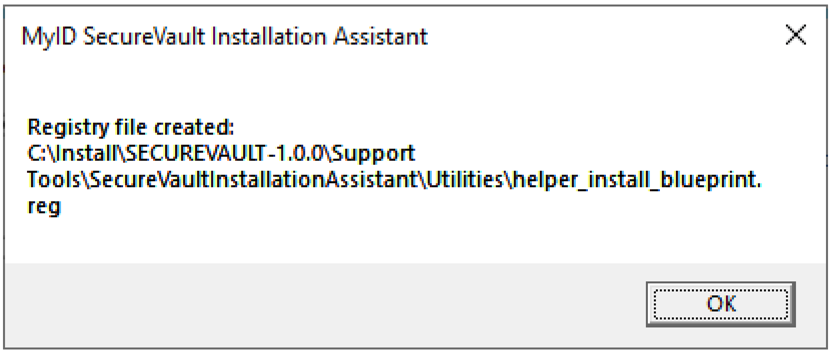
2.6.1 Exporting the registry file

To export the registry file:

1. On the installation options screen, click **Export Installation Registry File**.

   The installer exports the installation configuration to the `helper_install_blueprint.reg` file in the following folder:

   ```
   <install folder>\Support Tools\
   SecureVaultInstallationAssistant\Utilities\
   ```

   MyID SecureVault Installation Assistant

   Registry file created:
   C:\Install\SECUREVAULT-1.0.0\Support Tools\SecureVaultInstallationAssistant\Utilities\helper_install_blueprint.reg

2. Click **OK**.
3. Click **Cancel** to close the installer.

2.6.2 Populating the credentials in the registry file

For security reasons, the exported `helper_install_blueprint.reg` file strips out the usernames and passwords for the MyID SecureVault web service user and the SQL users (if you are using SQL Authentication).

To allow you to use these credentials when running the MyID SecureVault installer in automation mode, you can use the provided `SetupUsers.ps1` PowerShell script to insert these passwords into the registry file. This script uses DPAPI to encrypt the passwords; this means you must run the script on the machine on which you want to import the registry file, under the user account you will use to run the MyID SecureVault installation program.

To store the account passwords:

1. Log on using the Windows user account you will use to run the MyID SecureVault installation program, on the server on which you will run the MyID SecureVault installation program.
2. Open a Windows PowerShell command prompt.
3. Navigate to the following folder:

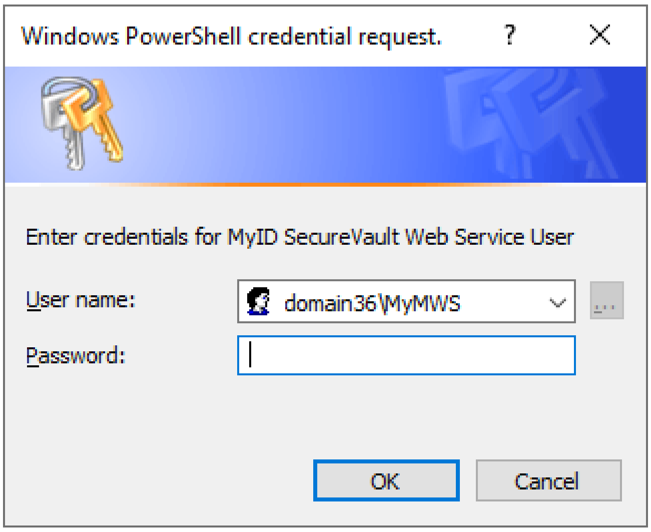
   ```
   <install folder>\Support Tools\SecureVaultInstallationAssistant\Utilities\
   ```
4. Make sure that there is no file called `SetupUsers.json` in the `Utilities` folder.

   This file is used for automating the `SetupUsers.ps1` script; see section *2.6.3, Automating the population of credentials in the registry file*. If this file exists, the script does not prompt for passwords, but instead loads the defaults from the file.

   A template `SetupUsers.json` file is provide by default; you can move or rename this file.
5. Run the following script:

   ```
   .\SetupUsers.ps1
   ```
6. Follow the on-screen prompts to provide the user credentials.

   The script prompts for each set of credentials:



Windows PowerShell credential request.

Enter credentials for MyID SecureVault Web Service User

User name: domain36\MyMWS

Password:

If the script does not prompt for passwords, check to make sure you do not have file called `SetupUsers.json` in the `Utilities` folder; if you do, remove it, and run the script again.

When the script is complete, it writes an updated registry file called:

```
helper_install.reg
```

to the following folder:

```
<install folder>\Support Tools\SecureVaultInstallationAssistant\
```

The MyID SecureVault installation program can now use this registry file to carry out the installation without any interaction.

2.6.3 Automating the population of credentials in the registry file

In some test environments, you may want to automate running the `SetupUsers.ps1` script so that no user interaction is required to populate the registry file with passwords. To do this, you can provide the passwords in plain text in a file called `SetupUsers.json` in the `utilities` folder.

Note: This file is short-lived. When you run the `SetupUsers.ps1` script, it extracts the passwords from this file, encrypts them, stores them in the `helper_install.reg` file, then (by default) deletes the `SetupUsers.json` file.

To automate the `SetupUsers.ps1` script:

1. Open the `SetupUsers.json` template file in the following folder:

   `<install folder>\Support Tools\SecureVaultInstallationAssistant\utilities\`

2. Edit the following:

```
[
        {
            "WSUser": "Domain\\WebServiceUser",
            "WSCred": "WebServiceUserCredential",
            "SecureVaultDBUser": "SecureVaultDBUser",
            "SecureVaultDBCred": "SecureVaultCDBred",
            "SecureVaultAuditDBUser": "SecureVaultAuditDBUser",
            "SecureVaultAuditDBCred": "SecureVaultAuditDBCred"
        }
]
```

   where:

   - `WSUser` – the domain and user for the MyID SecureVault web service user account.
   - `WSCred` – the password for the MyID SecureVault web service account in plain text.
   - `SecureVaultDBUser` – if you are using SQL authentication, the keys database user name.
   - `SecureVaultDBCred` – the database user password in plain text.
   - `SecureVaultAuditDBUser` – if you are using SQL authentication, the audit database user name.
   - `SecureVaultAuditDBCred` – the audit database user password in plain text.

   Note: Use double slashes for the slashes in the `domain\user` for each username; for example, for the `MYDOMAIN\MyUserName` account, use:

   `MYDOMAIN\\MyUserName`

   You do not need to include any items that you are not using for the current installation; for example, if you are using Windows authentication for database access, you do not need to provide database usernames and passwords, or if you are database server only, you do not need to provide the MyID SecureVault web service account details.

3. Save the file.

For example:

```
[
        {
            "WSUser": "MYDOMAIN\\MyWebServiceUser",
            "WSCred": "MyP455w0rd",
            "SecureVaultDBUser": "MyDatabaseUser",
            "SecureVaultDBCred": "MyP455w0rd",
            "SecureVaultAuditDBUser": "MyDatabaseUser",
            "SecureVaultAuditDBCred": "MyP455w0rd"
        }
]
```

You can now run the `SetupUsers.ps1` PowerShell script to insert these credentials into the registry file.

Important: By default, the `SetupUsers.ps1` script deletes the `SetupUsers.json` file on completion. If you want to retain the file, you can run the script with the following parameter:

```
.\SetupUsers.ps1 -KeepJsonFile $True
```

Note, however, that the `SetupUsers.json` file is *always* deleted when you run the MyID SecureVault installation program. If you want to retain the information in this file, make sure that you make a secure backup before you run the script; for security reasons, you are not recommended to leave this file with plaintext passwords freely available on the server for longer than is necessary.

2.6.4 Running the installation in automation mode

To run the MyID SecureVault installation program in automation mode:

1. Open a Windows PowerShell prompt, and navigate to the following folder:

   ```
   <install folder>\Support Tools\SecureVaultInstallationAssistant\
   ```

2. Run the following PowerShell script:

   ```
   .\SecureVaultInstallationAssistant.ps1 -Automation
   ```

When you run the MyID SecureVault installation program in automation mode, it loads the contents of the `helper_install.reg` file into the registry, including the encrypted passwords you added to the file, then starts at the first screen, and automatically moves through each screen without user interaction until it completes the installation of MyID SecureVault.

If an error occurs and the MyID SecureVault installation program stops, the screen on which the error occurred remains open.

Note: You can also run the installation in headless mode, where the installer carries out the same process but does not display any dialogs:

```
.\SecureVaultInstallationAssistant.ps1 -Automation -Headless
```

2.7 Configuring the database connection manually

You are recommended to set up and install the MyID SecureVault databases before you install the web services, or at the same time as you install the web services. However, if you cannot create your databases until later, you can skip the stage in the web service installation process that sets up the connection strings in the web service.

Once you have created and installed your database, you must configure the web service manually with the connection string.

Important: You must have the appropriate knowledge to create your own database connection strings. Creating database connection strings is beyond the scope of this document. If you cannot create your own database connection strings, you are recommended to create and set up the MyID SecureVault databases before you run the installation program, and let the installation program configure the web service with the appropriate connection strings.

To configure the database connection strings for the web service:

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:

   `C:\Program Files\Intercede\SecureVault\SecureVault`

2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.

3. Locate the placeholder text for the database connection strings:

   ```
   "ConnectionStrings": {
       "SecureVaultDatabase": "!SECUREVAULT_CONNECTION_STRING!",
       "SecureVaultAuditDatabase": "!SECUREVAULTAUDIT_CONNECTION_STRING!"
   },
   ```

4. Replace the following placeholders:

   - `!SECUREVAULT_CONNECTION_STRING!` – replace with the database connection string for the MyID SecureVault database.
   - `!SECUREVAULTAUDIT_CONNECTION_STRING!` – replace with the database connection string for the MyID SecureVault audit database.

5. Save the file.

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

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

For example, for Windows authentication, your database connection strings may look similar to the following:

```
"ConnectionStrings": {
    "SecureVaultDatabase": "Integrated Security=SSPI; Persist Security
Info=False; Initial Catalog=SecureVault; Data Source=VINF2K22DC01; Connection
Timeout=30; TrustServerCertificate=true;",
    "SecureVaultAuditDatabase": "Integrated Security=SSPI; Persist Security
Info=False; Initial Catalog=SecureVaultAudit; Data Source=VINF2K22DC01;
Connection Timeout=30; TrustServerCertificate=true;"
},
```

For SQL authentication, your database connection strings may look similar to the following:

```
"ConnectionStrings": {
    "SecureVaultDatabase": "Initial Catalog=SecureVault; Data
Source=VINF2K22DC01,1433; User Id=SecureVault;
PasswordDPAPI=AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA4OX6vAiBekiCRoyFObW5GQAAAAACAAA
AAAADZgAAwAAAABAAAAD3QHmtPVf6d3w4LPPkVrWuAAAAAASAAACgAAAAEAAAAIoFjOcdnFTkQWGh
IQUJb3QQAAAA+bftMtIjhnuv3yc3VeoMVhQAAACLU5Y6IIoSw85uhVvelOCYHnetjg==; Persist
Security Info=True; Connection Timeout=30; TrustServerCertificate=true;",
    "SecureVaultAuditDatabase": "Initial Catalog=SecureVaultAudit; Data
Source=VINF2K22DC01,1433; User Id=SecureVault;
PasswordDPAPI=AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA4OX6vAiBekiCRoyFObW5GQAAAAACAAA
AAAADZgAAwAAAABAAAADXjdnJGiwJzU7PnhErc3EwAAAAAASAAACgAAAAEAAAABpGRz2FxWGCl0WR
ER8OrrcQAAAAIA1LmeaQmPPoWRk+/p59ZhQAAACoVJrCISzXlavXalASd3y+Gf0bWg==; Persist
Security Info=True; Connection Timeout=30; TrustServerCertificate=true;"
},
```

Important: For SQL Authentication, you must encrypt the password and store it in the `PasswordDPAPI` parameter in the database connection strings; see section *2.7.1*, *Encrypting the password*.

2.7.1 Encrypting the password

If you are using SQL Authentication, you must encrypt the password before you store it in the configuration file for the web service.

MyID SecureVault supports DPAPI encryption for database passwords. The encrypted text can be decrypted only on the server on which it was encrypted, and only by the user who encrypted it.

You can use the following PowerShell script to encrypt the password:

```
Add-Type -AssemblyName System.Security
if ($args.count -ne 1)
{
  write-host "Usage: DPAPIEncrypt <text to encrypt>"
  exit
}
$clearText = $args[0]
$encbytes = [System.Security.Cryptography.ProtectedData]::Protect
([System.Text.Encoding]::UTF8.GetBytes($clearText), $null,
[System.Security.Cryptography.DataProtectionScope]::CurrentUser)
$b64 = [Convert]::ToBase64String($encbytes)
return $b64
```

If you are installing MyID SecureVault on the same web server as MyID CMS, you can find a copy of this script in the following folder:

`C:\Program Files\Intercede\MyID\web.oauth2\DPAPIEncrypt.ps1`

If you do not already have a copy of this script, save the above script as a file on your web server called `DPAPIEncrypt.ps1`.

Note: Make sure the script looks correct when you save it. Copying from a PDF file may cause issues with line breaks.

To encrypt the password:

1. Log on to your MyID SecureVault server as the MyID SecureVault web service user.

   Note: It is important that you use this account to encrypt the secret, as no other accounts can decrypt the password to use it.

2. Open a PowerShell command prompt.

3. Navigate to the folder where you saved the script, and run the following:

   `.\DPAPIEncrypt.ps1 <password>`

   where `<password>` is the password you want to encrypt.

   For example:

   `.\DPAPIEncrypt.ps1 MyP455w0rd`

   The script returns the DPAPI-encrypted password.

   ```
   PS C:\Scripts> .\DPAPIEncrypt.ps1 MyP455w0rd
   AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA5Zug2cAEmUuXWSwIx1wQOgAAAAACAAAAAAADZgAAw
   AAAABAAAACgI66ZWoHnyGUvL5ID3x9vAAAAAASAAACgAAAAEAAAAKXDjn/LHWJ9PmD9S+mDNJ
   cQAAAAMtlu/Ndt7RGweQJs0P+bBRQAAABIbOemw2Y6PN4lV4q46vlWinFgTg==
   PS C:\Scripts>
   ```

2.8 Uninstalling MyID SecureVault

Important: The MyID SecureVault uninstallation process requires PowerShell scripts that are provided as part of the MyID SecureVault installation package. If you move or delete the installation folder, you will be unable to uninstall MyID SecureVault using the Windows Control Panel **Programs and Features** option; when it is unable to locate the scripts, the uninstallation process displays an error. You are strongly recommended to retain the MyID SecureVault installation folder in the location from which you originally ran the installation program; this may influence your choice of folder from which to install MyID SecureVault .

You must use the **Programs and Features** option in the Windows Control panel to uninstall MyID SecureVault; due to an issue with Windows, you cannot use the Windows Apps & Features screen to uninstall MyID SecureVault.

2.8.1 Completely removing MyID SecureVault

Using the **Programs and Features** option in the Windows Control panel to uninstall MyID SecureVault removes the web service from IIS and deletes the `C:\Program Files\Intercede\SecureVault` folder, including the `appsettings.Production.json` file if you have created it.

The uninstallation process does not remove the databases. You must remove the databases manually.

The registry also retains the settings you provided when installing MyID SecureVault; this allows you to reinstall the software more quickly. You can remove these setting manually using the Registry Editor; the settings are stored in the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Intercede\Edefice\SecureVaultInstallation
```

3 Setting up authentication for the MyID SecureVault web service

The MyID SecureVault web service supports the following secure methods of authentication:

- OAuth2 authentication through the MyID CMS authentication server.

  See section *3.1*, *Setting up OAuth2 authentication*.
- Two-way TLS authentication using a client authentication certificate.

  See section *3.2*, *Setting up two-way TLS authentication*.

3.1 Setting up OAuth2 authentication

You can configure the MyID SecureVault server to accept a bearer token from the MyID CMS web.oauth2 server. You can use this bearer token to call the API, or to configure MyID CMS to integrate with MyID SecureVault through the **External Systems** workflow; see section *A.2*, *Setting up the MyID SecureVault external system*.

3.1.1 Configuring the web.oauth2 server

To configure the MyID CMS web.oauth2 server to create access tokens for the MyID SecureVault web service, you must:

- Create a client secret to be used to secure the connection.
- Add scopes for creating and recovering keys.

  There are separate scopes for the following operations:

  - `myid.securevault.create` – add keys to MyID SecureVault.
  - `myid.securevault.recover` – recover keys from MyID SecureVault.
  - `myid.securevault.biometric.write` – add, update, and delete biometric data in MyID SecureVault.
  - `myid.securevault.biometric.read` – search for and retrieve biometric data from MyID SecureVault.
- Add a custom API resource to contain these scopes.
- Add a custom client to use to connect to the web.oauth2 server.

3.1.1.1 Creating a client secret

You must produce a client secret and generate a Base64-encoded SHA-256 hash of this secret.

The *Server-to-server authentication* chapter of the *MyID Core API* guide contains information on creating a shared secret; alternatively, you can use the provided `GenClientSecret.ps1` PowerShell script to create a new secret and generate a Base64-encoded SHA-256 hash.

To generate a client secret and hash:

1. On the MyID CMS web server, open a PowerShell command prompt.
2. Navigate to the web.oauth2 folder.

   By default, this is:

   ```
   C:\Program Files\Intercede\MyID\web.oauth2\
   ```
3. Run the following:

   ```
   .\GenClientSecret.ps1
   ```

   The script displays its output on screen. For example:

   ```
   client secret: 036f2ed5-64c5-42d0-b57d-911e8950c568
   ```

   ```
   SHA256+base64: dBjzRnlYJqbQys+ZWoPROGISw7DGkSyYF8FJ3UDp3Aw=
   ```
4. Take a note of the output.

You need the first output value (`client secret`) when you set up the external system in MyID CMS or when authenticating to the API; see section *A.2*, *Setting up the MyID SecureVault external system* and section *5.2*, *Authenticating to the API*.

You need the second output value (`SHA256+base64`) when you create the custom client for MyID SecureVault; see section *3.1.1.4*, *Adding the MyID SecureVault custom client*.

3.1.1.2 Adding scopes for creating and recovering keys

To add the custom scopes:

1. On the MyID CMS web server, navigate to web.oauth2 folder. By default, this is:

   ```
   C:\Program Files\Intercede\MyID\web.oauth2\
   ```
2. If the `CustomScopes` subfolder does not exist, create it.
3. In the `CustomScopes` folder, create a text file called:

   ```
   securevaultcreate.json
   ```

   with the following content:

   ```
   {
       "Name":  "myid.securevault.create",
       "UserClaims":  [
       ]
   }
   ```

   This scope gives you the ability to add keys to MyID SecureVault.

4. In the `CustomScopes` folder, create a text file called:

   `securevaultrecover.json`

   with the following content:

   ```
   {
        "Name":  "myid.securevault.recover",
        "UserClaims":  [
        ]
   }
   ```

   This scope gives you the ability to recover keys from MyID SecureVault.

5. In the `CustomScopes` folder, create a text file called:

   `securevaultbiometricwrite.json`

   with the following content:

   ```
   {
        "Name":  "myid.securevault.biometric.write",
        "UserClaims":  [
        ]
   }
   ```

   This scope gives you the ability to add, update, and delete biometric data in MyID SecureVault.

6. In the `CustomScopes` folder, create a text file called:

   `securevaultbiometricread.json`

   with the following content:

   ```
   {
        "Name":  "myid.securevault.biometric.read",
        "UserClaims":  [
        ]
   }
   ```

   This scope gives you the ability to search for and retrieve single items of biometric data in MyID SecureVault.

3.1.1.3 Adding the custom API resource

To add the custom API resource:

1. On the MyID CMS web server, navigate to web.oauth2 folder. By default, this is:

   `C:\Program Files\Intercede\MyID\web.oauth2\`

2. If the `CustomApiResources` subfolder does not exist, create it.

3. In the `CustomApiResources` folder, create a text file called:

   `securevault.json`

   with the following content:

   ```
   {
       "Name":  "myid.securevault",
       "DisplayName":  "MyID SecureVault",
       "Scopes":  [
           "myid.securevault.create",
           "myid.securevault.recover",
           "myid.securevault.biometric.write",
           "myid.securevault.biometric.read"
       ]
   }
   ```

3.1.1.4 Adding the MyID SecureVault custom client

To add the custom client:

1. On the MyID CMS web server, navigate to web.oauth2 folder. By default, this is:

   `C:\Program Files\Intercede\MyID\web.oauth2\`

2. If the `CustomClients` subfolder does not exist, create it.

3. In the `CustomClients` folder, create a text file called:

   `securevault.json`

   with the following content:

   ```
   {
       "ClientId": "myid.securevault",
       "ClientName": "SecureVault Service",
       "AccessTokenLifetime": "3600",
       "AllowedGrantTypes": [
           "client_credentials"
       ],
       "ClientSecrets": [
       {
           "Value": "<secret>"
       }
       ],
       "AllowedScopes": [
           "myid.securevault.create",
           "myid.securevault.recover",
           "myid.securevault.biometric.write",
           "myid.securevault.biometric.read"
       ],
       "RedirectUris":  [
           "<securevault url>"
       ]
   }
   ```

   where:

   - `<secret>` is the Base64-encoded SHA-256 hash of the client secret you generated; see section *3.1.1.1*, *Creating a client secret*.
   - `<securevault url>` is the URL of the MyID SecureVault web service; for example:

     `https://myserver.example.com/securevault`

3.1.1.5 Cross-origin resource sharing

Cross-origin resource sharing (CORS) defines a way for web applications on one domain to interact with resources on other domains.

If you are using OAuth2 to authorize calls to SecureVault, and the web.oauth2 web service is on a different web origin to the web origin on which SecureVault is installed, you must configure CORS to allow the web.oauth2 server to be called from the web domain on which SecureVault is running.

To allow CORS:

1. In a text editor, open the custom client configuration file you created.

   For example:

   `C:\Program Files\Intercede\MyID\web.oauth2\CustomClients\securevault.json`

2. In the client you set up for your external system, add the following:

   ```
   "AllowedCorsOrigins": [
   "<external origin>"
   ]
   ```

   where:

   - `<external origin>` is the web origin on which SecureVault is installed. You can add multiple origins if necessary.

     Note: Make sure you use an origin, and not an URL, when configuring CORS. For example: `https://myserver/` is an URL, while `https://myserver` is an origin.

For example:

```
{
    "ClientId": "myid.securevault",
    "ClientName": "SecureVault Service",
    "AccessTokenLifetime": "3600",
    "AllowedGrantTypes": [
        "client_credentials"
    ],
    "ClientSecrets": [
    {
        "Value": "kv31VP5z/oKS0QMMaIfZ2UrhmQOdgAPpXV/vaF1cymk="
    }
    ],
    "AllowedScopes": [
        "myid.securevault.create",
        "myid.securevault.recover",
        "myid.securevault.biometric.write",
        "myid.securevault.biometric.read"
    ],
    "RedirectUris":  [
        "https://myserver.example.com/securevault"
    ],
    "AllowedCorsOrigins": [
        "http://myserver.example.com"
    ]
}
```

3. Save the configuration file.

3.1.1.6 Refreshing the authentication server settings

Once you have made your changes to the web.oauth2 server settings, you must refresh the application pool to ensure that all the systems are using the latest settings.

To refresh the server settings:

1. Recycle the web service app pool:
    a. On the MyID web server, in Internet Information Services (IIS) Manager, select **Application Pools**.
    b. Right-click the **myid.web.oauth2.pool** application pool, then from the pop-up menu click **Recycle**.

   This ensures that the web service has picked up the changes to the configuration file.
2. Check that the web.oauth2 server is still operational by logging on to the MyID Operator Client.

   Application setting JSON files are sensitive to such issues as comma placement; if the format of the file is not correct, the web service cannot load the file and will not operate, which may result in an error similar to:

   ```
   HTTP Error 500.30 - ANCM In-Process Start Failure
   ```

3.1.2 Configuring the MyID SecureVault server

You must configure the MyID SecureVault server to accept authentication using a bearer token from the authentication server.

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:

   `C:\Program Files\Intercede\SecureVault\SecureVault`

2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.

3. Edit the `Auth` section to contain the following:

```
"Auth": {
    "Bearer": {
        "AuthServerUrl": "<auth server>",
        "AllowUnsafeHttp": false,
        "ValidateIssuer": true,
        "Audience": "myid.securevault"
    },
    "AnonymousAccess": false,
    "AllowUnsafeHttp": false
},
```

   where:

   - `<auth server>` is the URL of your MyID CMS authentication server; for example:

     `https://myserver.example.com/web.oauth2`

4. Save the file.

5. Recycle the web service app pool:

   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.

   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.

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

Important: You have now configured the authentication for the MyID SecureVault web service, but the configuration of the web service is not complete. If you attempt to connect to the MyID SecureVault web service (through the MyID External Systems workflow, or through the API) the connection fails with an error similar to:

`VLT10001 - A protection error has occurred.`

This is because you have not yet configured the encryption settings. See section *4*, *Configuring MyID SecureVault* for details.

3.2 Setting up two-way TLS authentication

You can configure the MyID SecureVault server to accept a certificate for two-way TLS authentication. You can use this certificate to call the API, or to configure MyID CMS to integrate with MyID SecureVault through the **External Systems** workflow; see section *A.2*, *Setting up the MyID SecureVault external system*.

3.2.1 Generating a certificate

You must generate a certificate for the PC that is going to call MyID SecureVault; if you are integrating MyID CMS with MyID SecureVault, this is the MyID application server.

If you have an existing PKI infrastructure, you can use that to issue a client authentication certificate. Alternatively, you can generate a self-signed certificate.

You can use the following PowerShell script to generate a self-signed certificate:

```
$params = @{
    Type = 'Custom'
    Subject = 'CN=SecureVault Client'
    TextExtension = @('2.5.29.37={text}1.3.6.1.5.5.7.3.2')
    KeyUsage = 'DigitalSignature'
    KeyAlgorithm = 'RSA'
    KeyLength = 3072
    NotAfter = (Get-Date).AddYears(5)
    CertStoreLocation = 'Cert:\CurrentUser\My'
}
New-SelfSignedCertificate @params
```

Note: The certificate uses the default provider, which is the Microsoft Software Key Storage Provider. If you want to use an HSM instead, you can use the `Provider` parameter for the `New-SelfSignedCertificate` cmdlet to specify the HSM KSP. See the Microsoft documentation for the `New-SelfSignedCertificate` cmdlet for details.

To create a self-signed certificate:

1. On the PC that is going to call MyID SecureVault, log in as the user that is going to call the API.

   For example, on the MyID application server, log in as the MyID COM user.

2. Run the above PowerShell script.

3. Take a note of the `Thumbprint` and `Subject` output from the script.

   For example:

   ```
   Thumbprint                                Subject
   ----------                                -------
   34653DD64AD882F78CEE4C6520ABD83441BB4AA6  CN=SecureVault Client
   ```

   You need the `Thumbprint` to configure the application settings for the MyID SecureVault web service, and the `Subject` to configure the external system in MyID CMS.

3.2.2 Configuring MyID SecureVault with the details of the certificate

Once you have generated the certificate for the PC and user account that are going to call the MyID SecureVault API, you must provide the details of this certificate to the MyID SecureVault web service.

To configure the certificate details:

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:

   `C:\Program Files\Intercede\SecureVault\SecureVault`

2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.

3. Edit the `Auth` section to contain the following:

   ```
   "Auth":{
     "ClientCertificate":{
       "RevocationMode":"NoCheck",
       "AllowedCertificates":[
         {
           "Thumbprint":"<thumbprint>",
           "Scopes":[
             "myid.securevault.create",
             "myid.securevault.recover",
             "myid.securevault.biometric.write",
             "myid.securevault.biometric.read"
           ]
         }
       ]
     }
   }
   ```

   where:

   - `<thumbprint>` is the thumbprint of the client authentication certificate.

Note: You can include multiple certificates in the `AllowedCertificates` array if you need to call the MyID SecureVault API from multiple servers or user accounts.

You must include `myid.securevault.create` and `myid.securevault.recover` in the `Scopes` array for each certificate. If you want to store and recover biometric data, you must also include the `myid.securevault.biometric.write` and `myid.securevault.biometric.read` scopes.

For example:

```
"Auth":{
  "ClientCertificate":{
    "RevocationMode":"NoCheck",
    "AllowedCertificates":[
      {
        "Thumbprint":"34653DD64AD882F78CEE4C6520ABD83441BB4AA6",
        "Scopes":[
          "myid.securevault.create",
          "myid.securevault.recover",
          "myid.securevault.biometric.write",
          "myid.securevault.biometric.read"
        ]
      },
      {
        "Thumbprint":"1B28ACFA999DF9CF2915A02066644592BCC7E126",
        "Scopes":[
          "myid.securevault.create",
          "myid.securevault.recover",
          "myid.securevault.biometric.write",
          "myid.securevault.biometric.read"
        ]
      }
    ]
  }
}
```

4. Recycle the web service app pool:
   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.

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

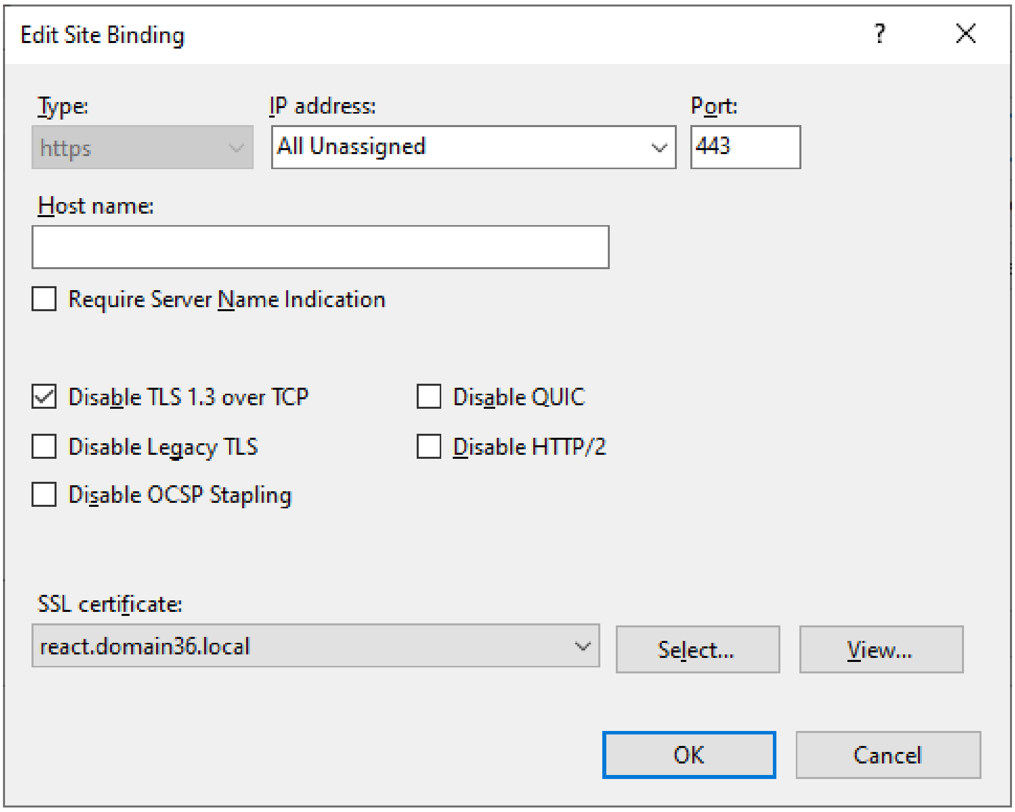
3.2.3 Configuring IIS for two-way TLS

You must configure IIS to allow two-way TLS.

1. Ensure TLS 1.3 is not enabled for the web server:
   a. In Internet Information Services (IIS) Manager, in the **Connections** pane, expand the server name, then **Sites**, then select the website used for MyID SecureVault; by default, this is **Default Web Site**.
   b. Right-click the website, then from the pop-up menu select **Edit Bindings**.
   c. In the Site Bindings dialog, select **https**.

d. Click **Edit**.

e. Select the **Disable TLS 1.3 over TCP** option.



Edit Site Binding

Type: https

IP address: All Unassigned

Port: 443

Host name:

□ Require Server Name Indication

☑ Disable TLS 1.3 over TCP

□ Disable QUIC

□ Disable Legacy TLS

□ Disable HTTP/2

□ Disable OCSP Stapling

SSL certificate: react.domain36.local

f. Click **OK**, then click **Close**.

Note: You *can* enable TLS 1.3 on Windows Server 2022; however, this requires additional configuration. See section *3.2.5*, *Using TLS 1.3*.

2. Enable two-way TLS for the SecureVault application:

   a. Expand the website used for MyID SecureVault (by default, **Default Web Site**) and select **SecureVault**.

   b. Double-click **SSL Settings**.

   c. Select the **Require SSL** option.

   d. For the **Client certificates** option, select **Accept** or **Require**.

      You can use **Accept** as well as **Require** because the MyID SecureVault web service configuration specifies the accepted client certificates.

   e. Click **Apply**.

3. Recycle the web service app pool:

   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.

   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.

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

3.2.4 Trusting the client certificate on the MyID SecureVault server

The MyID SecureVault server must trust the client certificate you are using for two-way TLS.

If you issued your certificate from a certificate authority, make sure that the root CA is trusted on your MyID SecureVault server.

If you issued a self-signed certificate, you must export the certificate from your Personal store, then import the certificate to the **Local Computer\Trusted Root Certification Authorities** store on the MyID SecureVault server.

3.2.5 Using TLS 1.3

By default, MyID SecureVault is installed to the Default Web Site in IIS; if you attempt to set up MyID SecureVault for TLS 1.3, this may cause issues with other web applications that share the same IIS website. To use TLS 1.3 with MyID SecureVault, you must move the SecureVault virtual directory into its own website.

Create a new website within IIS, then create a SecureVault virtual directory within that website. Copy all the settings from the originally-installed SecureVault virtual directory to the new virtual directory you have created.

Once you have done this, you must modify the https port bindings to enable client certificate negotiation. You cannot do this through the IIS user interface, but you can use a PowerShell script instead.

To enable the use of TLS 1.3 for MyID SecureVault:

1. Ensure TLS 1.3 is enabled for the web server:
   a. In Internet Information Services (IIS) Manager, in the **Connections** pane, expand the server name, then **Sites**, then select the new website you created to use for MyID SecureVault.
   b. Right-click the website, then from the pop-up menu select **Edit Bindings**.
   c. In the Site Bindings dialog, select **https**.
   d. Click **Edit**.
   e. Make sure the **Disable TLS 1.3 over TCP** option is not selected.
   f. Click **OK**, then click **Close**.
2. Take a note of the server TLS certificate thumbprint:
   a. In IIS, select the server on which MyID SecureVault is installed, then double-click **Server Certificates**.
   b. Double-click the certificate used for TLS authentication.
   c. Select the **Details** tab.

      You can view the **Thumbprint** field here.

3. Run the following PowerShell script, substituting in the thumbprint in the `serverTlsCertThumbprint` variable:

```
$port = 443
$securevault_port = 1443
$serverTlsCertThumbprint = 'acac5d7711027dbf2af4b0cbff6256b6e759954b'
$iisBindingAppId = New-Guid
$securevault_iisBindingAppId = New-Guid
Start-Process -FilePath "netsh" -ArgumentList @("http", "del", -join
("sslcert", " 0.0.0.0:", $port)) -Wait
Start-Process -FilePath "netsh" -ArgumentList @("http", "add", "sslcert",
-join("ipport=", "0.0.0.0:", $port), -join("certhash=",
$serverTlsCertThumbprint), -join("appid={", $iisBindingAppId, "}"),
"certstorename=MY", "clientcertnegotiation=Disable") -Wait
Start-Process -FilePath "netsh" -ArgumentList @("http", "del", -join
("sslcert", " 0.0.0.0:", $securevault_port)) -Wait
Start-Process -FilePath "netsh" -ArgumentList @("http", "add", "sslcert",
-join("ipport=", "0.0.0.0:", $securevault_port), -join("certhash=",
$serverTlsCertThumbprint), -join("appid={", $securevault_iisBindingAppId,
"}"), "certstorename=MY", "clientcertnegotiation=Enable") -Wait
```

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

3.2.6 Using a load balancer or reverse proxy

If you are using a load balancer or reverse proxy, you must set up the certificate forwarding details for the MyID SecureVault web service. The load balancer or reverse proxy terminates the two-way TLS connection then passes the certificate used by the client to authenticate to the web service as an http header; you must configure the MyID SecureVault web service to determine the name of the header that contains the certificate, and whether the certificate is hex encoded or URL encoded; the encoding used depends on which load balancer or reverse proxy you are using.

To configure the certificate forwarding details:

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:

   `C:\Program Files\Intercede\SecureVault\SecureVault`

2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.

3. Edit the `Auth` section to contain the following additional settings:

```
"CertificateForwarding": {
  "CertificateHeader": "<CertificateHeader>",
  "HeaderConverterMethod": "<HeaderConverterMethod>"
}
```

where:

- `<CertificateHeader>` is the name of the request header containing the certificate; for example:

  `Certificate`

- `<HeaderConverterMethod>` is one of the following:
  - `HexEncoded` – the certificate uses hex encoding.
  - `UrlEncoded` – the certificate uses URL encoding.

For example:

```
"Auth":{
  "ClientCertificate":{
    "RevocationMode":"NoCheck",
    "AllowedCertificates":[
      {
        "Thumbprint":"34653DD64AD882F78CEE4C6520ABD83441BB4AA6",
        "Scopes":[
          "myid.securevault.create",
          "myid.securevault.recover",
          "myid.securevault.biometric.write",
          "myid.securevault.biometric.read"
        ]
      },
      {
        "Thumbprint":"1B28ACFA999DF9CF2915A02066644592BCC7E126",
        "Scopes":[
          "myid.securevault.create",
          "myid.securevault.recover",
          "myid.securevault.biometric.write",
          "myid.securevault.biometric.read"
        ]
      }
    ]
  },
  "CertificateForwarding": {
    "CertificateHeader": "Certificate",
    "HeaderConverterMethod": "HexEncoded"
  }
}
```

4. Save the file.

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

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

4 Configuring MyID SecureVault

You must configure MyID SecureVault to specify whether you are using a self-signed certificate or an HSM to generate and store keys or encrypt data.

You can set up your system without an HSM, then subsequently add support for an HSM; you can still use the certificate to recover keys and data that you generated and stored before setting up the HSM.

This section contains the following information:

- Setting up MyID SecureVault to use a self-signed certificate.

  See section *4.1*, *Configuring MyID SecureVault for use without an HSM*.
- Setting up MyID SecureVault to use an HSM.

  See section *4.2*, *Configuring MyID SecureVault for use with an HSM*.

4.1 Configuring MyID SecureVault for use without an HSM

You can use MyID SecureVault without an HSM. This mode uses a self-signed encryption certificate to store encrypted end entity private keys in CMS (RFC 5652) format) and uses the Windows provider (certified to FIPS140-2 level 1) for key generation of end entity private keys.

You must generate a self-signed certificate, then provide its details to the MyID SecureVault web service.

4.1.1 Generating a self-signed certificate

You can use the following PowerShell script to generate a self-signed certificate:

```
$params = @{
    Type = 'Custom'
    Subject = 'CN=SecureVault Encryption 1'
    KeyUsage = 'DataEncipherment'
    KeyAlgorithm = 'RSA'
    KeyLength = 4096
    NotAfter = (Get-Date).AddYears(20)
    CertStoreLocation = 'Cert:\CurrentUser\My'
    Provider='Microsoft Software Key Storage Provider'
}
New-SelfSignedCertificate @params
```

1. Log on to the MyID SecureVault server as the MyID SecureVault web service user.
2. Run the above PowerShell script.
3. Take a note of the `Thumbprint` and `Subject` output from the script.

   For example:

   ```
   Thumbprint                                Subject
   ----------                                -------
   7524E011C0CC61A3E492488FA9119C410129D607  CN=SecureVault Encryption 1
   ```

The certificate is generated and added to the Personal store for the MyID SecureVault web service user.

Important: You must back up this encryption certificate – export it to a PFX file then store it in a secure location. You must add this certificate to the Personal certificate store of the MyID SecureVault web service user on each MyID SecureVault server. If you need to build another server, you must import this certificate on the new server to allow it to access the end entity private keys encrypted with the certificate.

4.1.2 Configuring the MyID SecureVault web service with the certificate

You must add the thumbprint and subject DN of the certificate to the MyID SecureVault configuration file:

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:

   `C:\Program Files\Intercede\SecureVault\SecureVault`

2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.

3. Edit the `MicrosoftEncryptionOptions` section as follows:

   ```
   "MicrosoftEncryptionOptions": {
       "Thumbprint": "<thumbprint>",
       "SubjectDN": "<subject>"
   },
   ```

   where:

   - `<thumbprint>` is the thumbprint of the certificate.
   - `<subject>` is the subject DN of the certificate.

   Note: You do not have to include both the thumbprint and the subject DN; if you include either, MyID SecureVault finds the relevant certificate.

   The PowerShell script listed in section *4.1.1*, *Generating a self-signed certificate* outputs both of these values.

   For example:

   ```
   "MicrosoftEncryptionOptions": {
       "Thumbprint": "7524E011C0CC61A3E492488FA9119C410129D607",
       "SubjectDN": "CN=SecureVault Encryption 1"
   },
   ```

4. Remove the `HsmKeyProtectorOptions` section.

   This section is used only for systems that use an HSM.

5. Save the file.

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

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

4.2 Configuring MyID SecureVault for use with an HSM

When you configure MyID SecureVault to use a Hardware Security Module (HSM), end-entity private keys are protected by an HSM-resident AES256 key; when you generate end-entity private keys, these are generated on the HSM.

MyID SecureVault supports the following HSMs:

- Entrust nShield Connect
- Entrust nShield Solo
- Thales Luna

You must install and configure your HSM, including installing the HSM client software on the MyID SecureVault server, before you configure MyID SecureVault.

Note: The generation of private keys requires an HSM partition capable of generating private keys.

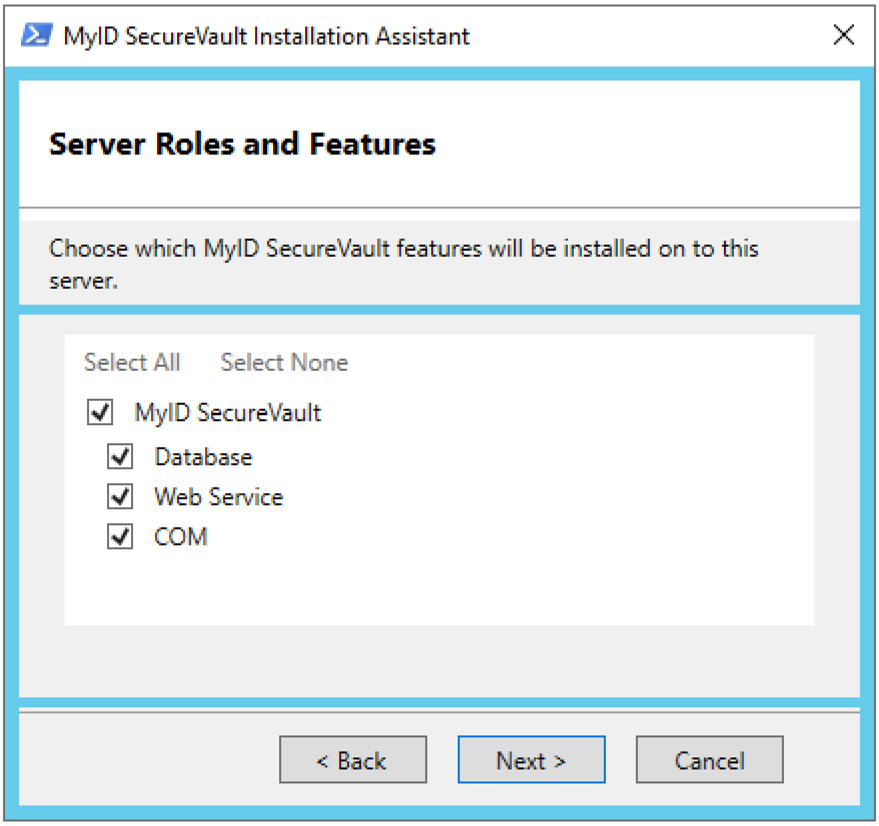
You can set up your system without an HSM (see section *4.1*, *Configuring MyID SecureVault for use without an HSM*), then subsequently add support for an HSM; you can still use the certificate to recover keys that you generated and stored before setting up the HSM. Do not remove the configuration you set up for certificate-based key generation; if the configuration for HSM key generation is present, MyID SecureVault automatically prefers to use the HSM.

Note: You cannot modify an installation to add the COM option, which is required for HSM support. If you intend to use an HSM with MyID SecureVault at any point, you must install MyID SecureVault with HSM support.

4.2.1 Installing MyID SecureVault with HSM support

If you intend to use an HSM with MyID SecureVault, when you install MyID SecureVault, you must either:

- Select the **COM** option on the Server Roles and Features screen of the installation program.



MyID SecureVault Installation Assistant

Server Roles and Features

Choose which MyID SecureVault features will be installed on to this server.

Select All Select None

☑ MyID SecureVault
☑ Database
☑ Web Service
☑ COM

  Install these components on the same server as the web service.

  or:

- Install MyID SecureVault on a MyID CMS application server.

  When you select the COM option, the installation program installs COM components that are used to access the HSM. These components are not required on a MyID CMS application server, as the components are already installed as part of the MyID CMS installation, and MyID SecureVault can make use of them.

Important: If you are installing MyID SecureVault onto the same server as the MyID CMS application server, *do not* install the COM components. If you install the COM components on a MyID application server, and then subsequently uninstall MyID SecureVault, you may damage your MyID CMS installation.

4.2.2 Configuring MyID SecureVault to use an HSM with MyID CMS

If you have MyID SecureVault installed on the same server as the MyID CMS application server, MyID SecureVault uses the configuration that you set up for your HSM within MyID CMS.

Note: This applies to the primary partition on the HSM only; as MyID CMS does not support alternative key partitions, if you want to use these additional partitions, you must configure MyID SecureVault with their details. See section *4.2.5*, *Configuring alternative key partitions*.

See the MyID CMS integration guide for your HSM for details of installing and configuring your HSM.

If you are using an HSM that requires a PIN, make sure you save the HSM PIN; see the *Using GenMaster* and *Setting the HSM PIN* sections of the *Installation and Configuration Guide* in the MyID CMS documentation set.

Important: When MyID CMS stores the PIN for the HSM, it uses the MyID COM+ user. You must also set the PIN for the HSM using the SetHSMPIN utility as the MyID SecureVault web service user.

4.2.3 Configuring MyID SecureVault to use an HSM as a standalone server

If you do not have MyID SecureVault installed on the same server as the MyID CMS application server, but have a standalone installation, you must carry out additional configuration to enable HSM support.

> **Warning:** Do not carry out this procedure on a MyID CMS application server, or you may damage your MyID CMS installation.

To generate a database key and configure the registry:

1. On the MyID SecureVault server, log in as an account with local administrator permissions.
2. Open a Windows PowerShell command prompt.
3. Navigate to the following folder:

   ```
   <install folder>\Installer\SECUREVAULT-<version>\
   Scripts\AdditionalScripts
   ```

4. Run the following PowerShell script:

   ```
   .\GenerateDatabaseKey.ps1
   ```

   with the following parameters:

   - `-Generator` – The name of the crypto module to use.

     For a Thales Luna HSM:

     ```
     -Generator "LUNA Crypto Module"
     ```

     For an Entrust nShield HSM:

     ```
     -Generator "nCipher Crypto Module"
     ```

   - `-SerialNumber` – The serial number of the HSM partition, or the virtual slot serial number if you want to reference a virtual HSM slot that is configured for load balancing.
   - `-StandAlone` – Set to `1` to indicate that the MyID SecureVault server is separate to the MyID application server.
   - `-DatabaseKeyName` – The required name for the database key. If this is not present, a GUID is generated.
   - `-UseExistingKey` – Set to `1` if you want to use an existing key on the HSM .

     Note: Generation fails if you set this option to `0` and the key name specified in `-DatabaseKeyName` already exists on the HSM.
   - `-SavePIN` – Set to `1` to save the HSM PIN.

     Set this option if your HSM requires a PIN.
   - `-Pin` – The PIN for the HSM.

     Set this option if your HSM requires a PIN.
5. Once the script has completed, create a backup of the main HSM partition.

4.2.3.1 Setting the PIN for the main partition

If your HSM requires a PIN, you must use the `SetPin.ps1` script to set or change the PIN for the main partition.

Note: You must run this script to set the PIN even though you provided the PIN when running the `GenerateDatabaseKey.ps1` script above; this script writes to a different registry location, and requires different permissions to run.

To set the PIN:

1. On the MyID SecureVault server, log in as the MyID SecureVault web service user.
2. Open a Windows PowerShell command prompt.
3. Navigate to the following folder:
   ```
   <install folder>\Installer\SECUREVAULT-<version>\
   Scripts\AdditionalScripts
   ```
4. Run the following PowerShell script:
   ```
   .\SetPin.ps1 -Pin <PIN> -HsmKeyName <key>
   ```
   where:
   - `<PIN>` is the HSM PIN you want to save.
   - `<key>` is one of the following:
     - `LUNA` – used for Thales Luna HSMs.
     - `nCipher` – used for Entrust nShield HSMs.
5. Recycle the web service app pool:
   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.

   This ensures that the web service has picked up the changes.

Note: You can also use this script to set the PINs for transient partitions; see section *4.2.5.3, Setting the HSM PIN for alternative partitions*.

4.2.4 Configuring the MyID SecureVault web service to use an HSM

To configure the MyID SecureVault web service to use an HSM:

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:
   ```
   C:\Program Files\Intercede\SecureVault\SecureVault
   ```
2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.

3. Edit the `HsmKeyProtectorOptions` section as follows:

   ```
   "HsmKeyProtectorOptions": {
     "KeyProvider": "<KeyProvider>",
     "NumSessions": <NumSessions>,
     "ConcurrencyEnabled": <ConcurrencyEnabled>,
     "AltKeyProvider": []
   },
   ```

   where:

   - `<KeyProvider>` is the ProgID of the HSM key provider; for example:
     - `LUNAKeySrv.LUNAKeyServer.1` – used for Thales Luna HSMs.
     - `NCKeySrv.NCKeyServer.1` – used for Entrust nShield HSMs.

     If you specify a value for the `KeyProvider`, MyID SecureVault attempts to use an HSM; if you leave the `KeyProvider` property as an empty string `""`, MyID SecureVault uses software encryption instead; see section *4.1*, *Configuring MyID SecureVault for use without an HSM*.
   - `<NumSessions>` is the number of concurrent sessions to the HSM.

     See section *4.2.6*, *Considerations for setting the number of concurrent sessions*.
   - `<ConcurrencyEnabled>` is whether concurrent sessions are enabled; set this property to `1` to enable concurrent sessions.
   - `<AltKeyProvider>` is an array of alternative key providers; see section *4.2.5*, *Configuring alternative key partitions*.

     Note: You must include the `AltKeyProvider` property; if you do not want to specify alternative key providers, use an empty array `[]`.

   For example:

   ```
   "HsmKeyProtectorOptions": {
     "KeyProvider": "LUNAKeySrv.LUNAKeyServer.1",
     "NumSessions": 6,
     "ConcurrencyEnabled": 1,
     "AltKeyProvider": []
   },
   ```

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

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

4.2.5 Configuring alternative key partitions

For MyID SecureVault to recover asymmetric keys, the HSM must support asymmetric key export. Some HSM partitions may not support this feature; if the main HSM partition does not support asymmetric key export, you can configure alternative key partitions that *do* support asymmetric key export.

If your main HSM partition supports asymmetric key export, you do not need to configure any alternative key partitions.

You can configure MyID SecureVault to use the following alternative partitions:

- `KeygenTransient`

  If configured, this partition is used to generate transient keys.
- `BackUpKeygenTransient`

  If configured, this partition is used as a backup to the `KeygenTransient` partition. If the connection to the `KeygenTransient` partition fails, MyID SecureVault attempts to generate the transient key on this backup partition.

To configure an alternative key partition, you must:

- Add the serial number of the partition to the registry.
- Update the application settings configuration for the MyID SecureVault server.
- If you are using an HSM that requires a PIN, encrypt and store the PIN for the partition in the registry.

4.2.5.1 Adding the serial number of the alternative partition to the registry

For `KeygenTransient`, add the following key to the registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Intercede\Edefice\MasterCard\KeyGenTransient
```

Within this key, add the following String (`REG_SZ`) value:

- `SerialNumber` – set this to the serial number of the HSM partition being used for transient key generation.

For `BackupKeygenTransient`, add the following key to the registry:

```
HKEY_LOCAL_
MACHINE\SOFTWARE\Intercede\Edefice\MasterCard\BackupKeyGenTransient
```

Within this key, add the following String (`REG_SZ`) value:

- `SerialNumber` – set this to the serial number of the HSM partition being used for backup transient key generation.

4.2.5.2 Configuring the alternative partition in the MyID SecureVault web service

To set up the MyID SecureVault web service to use an alternative partition:

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:

   ```
   C:\Program Files\Intercede\SecureVault\SecureVault
   ```

2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.

3. Edit the `AltKeyProvider` array in the `HsmKeyProtectorOptions` section.

   Add an entry to the array for each alternative partition with the following details:

   - `UseAlternativePartition` – set to `true`.
   - `KeyProvider` – set to the ProgID of the key provider.
   - `NumSessions` – set to the number of concurrent sessions to the HSM.

     See section *4.2.6*, *Considerations for setting the number of concurrent sessions*.
   - `Type` – set to one of the following:
     - `KeyGenTransient`
     - `BackupKeyGenTransient`

   For example:

```
"AltKeyProvider":[
  {
    "UseAlternatePartition":true,
    "KeyProvider":"LUNAKeySrv.LUNAKeyServerAlt",
    "NumSessions":6,
    "Type":"KeyGenTransient"
  },
  {
    "UseAlternatePartition":true,
    "KeyProvider":"LUNAKeySrv.LUNAKeyServerAlt",
    "NumSessions":4,
    "Type":"BackupKeyGenTransient"
  }
]
```

4. Save the file.

5. Recycle the web service app pool:

   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.

   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.

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

4.2.5.3 Setting the HSM PIN for alternative partitions

You must set the PIN for each alternative partition.

1. On the MyID SecureVault server, log in as the MyID SecureVault web service user.

   Note: This account must have local administrator permissions.

2. Open a Windows PowerShell command prompt.
3. Navigate to the following folder:

   ```
   <install folder>\Installer\SECUREVAULT-<version>\
   Scripts\AdditionalScripts
   ```

4. Run the following PowerShell script:

   ```
   .\SetPin.ps1
   ```

   with the following parameters:

   - `-Pin` – set to the PIN for the partition.
   - `-HsmKeyName` – set to one of the following:
     - `KeyGenTransient`
     - `BackupKeyGenTransient`

5. Recycle the web service app pool:

   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.

   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.

   This ensures that the web service has picked up the changes.

4.2.6 Considerations for setting the number of concurrent sessions

When you call MyID SecureVault, the initial connection to the HSM may take some time. The initial connection authenticates a configurable number of sessions to the main partition, and also a number of sessions to the alternative partitions (if configured).

The time this connection takes depends on a variety of factors, including the speed of the HSM and the network latency between MyID SecureVault and the HSM.

If this initial connection takes too long, the client (for example, the MyID CMS application server) may reach a timeout before MyID SecureVault responds.

In this case, you may want to reduce the number of concurrent sessions to lower the initial connection time. To do this, change the `NumSessions` option for each partition in the `appsettings.Production.config` file for the MyID SecureVault web service; see section *4.2.4*, *Configuring the MyID SecureVault web service to use an HSM* and section *4.2.5.2*, *Configuring the alternative partition in the MyID SecureVault web service*.

There is a trade-off between retaining the number of concurrent sessions, which allows more concurrent operations at the cost of startup time, and reducing the number of concurrent sessions, which reduces the number of concurrent operations but improves the startup time. This balance depends on your environment and network.

If you are using MyID CMS to integrate with MyID SecureVault, you can test the startup time as follows:

1. Recycle the web service app pool:
   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.
2. In MyID Desktop, from the **Configuration** category, select **External Systems**.
3. Select the MyID SecureVault external system, click **Edit**, then click **Test Connection**.

This process makes an initial connection to the HSM. You can try adjusting the `NumSessions` values and check the effect it has on the startup time.

5 Using the MyID SecureVault API

MyID SecureVault provides a REST API that you can use to work with keys or data.

The following `Keys` endpoints are available:

- Import a key.

  `POST /api/Keys/import`

- Generate a new key.

  `POST /api/Keys/generate`

- Sign data with a key.

  `POST /api/Keys/{id}/sign`

- Recover a key.

  `POST /api/Keys/{id}/recover`

- Check that the MyID SecureVault web service is running.

  `POST /api/Keys/isAlive`

The following `Data` endpoints are available, allowing you to work with biometric data:

- Add a biometric data item to be encrypted in storage.

  `POST /api/Data/biometric`

- Retrieve a set of biometric data items.

  `GET /api/Data/biometric`

- Add a batch of biometric data items to be encrypted in storage.

  `POST /api/Data/batch/biometric`

- Update a batch of biometric data items in storage.

  `PATCH /api/Data/batch/biometric`

- Delete a batch of biometric data items from storage.

  `DELETE /api/Data/biometric`

- Retrieve a single biometric data item from storage.

  `GET /api/Data/biometric/{id}`

- Update a single biometric data item in storage.

  `PATCH /api/Data/biometric/{id}`

- Delete a single biometric data item from storage.

  `DELETE /api/Data/biometric/{id}`

Full details of these endpoints are available in the online API documentation; see section *5.1*, *Accessing the API documentation*.

You can authenticate to the API using two-way TLS or OAuth2 credentials; see section *5.2*, *Authenticating to the API*.

Once you have authenticated to the API, you can call the endpoints to carry out key or data operations; see section *5.3*, *Calling the API*.

5.1 Accessing the API documentation

The API documentation is provided on the API server at the following address:

`https://<myserver>/SecureVault/swagger/index.html`

where `<myserver>` is the name of the server hosting the MyID SecureVault web service.

If necessary, you can configure the API documentation to prevent access. You can also to configure the API documentation to allow you to use the documentation as a test harness, subject to the appropriate authentication.

To configure access the API documentation:

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:

   `C:\Program Files\Intercede\SecureVault\SecureVault`

2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.

3. Edit the following section:

   ```
   "Swagger": {
     "GenerateDocumentation": true,
     "AllowTestFromDocumentation": true
   },
   ```

   Set the following options:

   - `GenerateDocumentation` – set this to `true` to produce API documentation at the Swagger URL for the MyID SecureVault web service, or `false` to prevent access to the documentation and display a 404 not found error for the API documentation URL.

     The default is `true`.

   - `AllowTestFromDocumentation` – set this to `true` to allow anyone with access to the API documentation to call each endpoint from the API documentation page, or `false` to prevent authentication using OAuth2 to access to this feature.

     Note: Setting this option to `false` does not affect access to this feature if your MyID SecureVault web service is configured for two-way TLS.

     If you enable this feature, you must still provide the appropriate authentication (two-way TLS or OAuth2) before you can call the API.

     The default is `true`.

     Note: If the web.oauth2 web service is on a different web origin to the web origin on which SecureVault is installed, you must configure CORS; see section *3.1.1.5*, *Cross-origin resource sharing*.

4. Recycle the web service app pool:
   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.

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

5.2 Authenticating to the API

The method you use to authenticate to the API depends on the authentication you have configured for the MyID SecureVault web service:

- OAuth2

  For OAuth2 authentication, you need the following information:

  - The OAuth2 grant type:

    `client_credentials`

  - The client ID you set up; for example:

    `myid.securevault`

  - The scopes you set up:

    - `myid.securevault.create`
    - `myid.securevault.recover`
    - `myid.securevault.biometric.write`
    - `myid.securevault.biometric.read`

  - The client secret you set up; for example:

    `036f2ed5-64c5-42d0-b57d-911e8950c568`

  For information on configuring MyID SecureVault to use OAuth2, see section *3.1*, *Setting up OAuth2 authentication*.

- Two-way TLS

  For two-way TLS authentication, you must have the client certificate installed on your PC.

  For information on configuring MyID SecureVault to use two-way TLS, see section *3.2*, *Setting up two-way TLS authentication*.

This section contains examples of authenticating to the API from the API documentation and from PowerShell. The same principles apply if you are authenticating from another platform or using a different language.

5.2.1 Authenticating to the API from the API documentation

You can call the API endpoints from the API documentation; see section *5.1*, *Accessing the API documentation*. You can authenticate in your browser using OAuth2 or two-way TLS.

5.2.1.1 Authenticating using OAuth2

To authenticate using OAuth2 from the API documentation:

1. Open a web browser and navigate to the MyID SecureVault API documentation:

   `https://<myserver>/SecureVault/swagger/index.html`

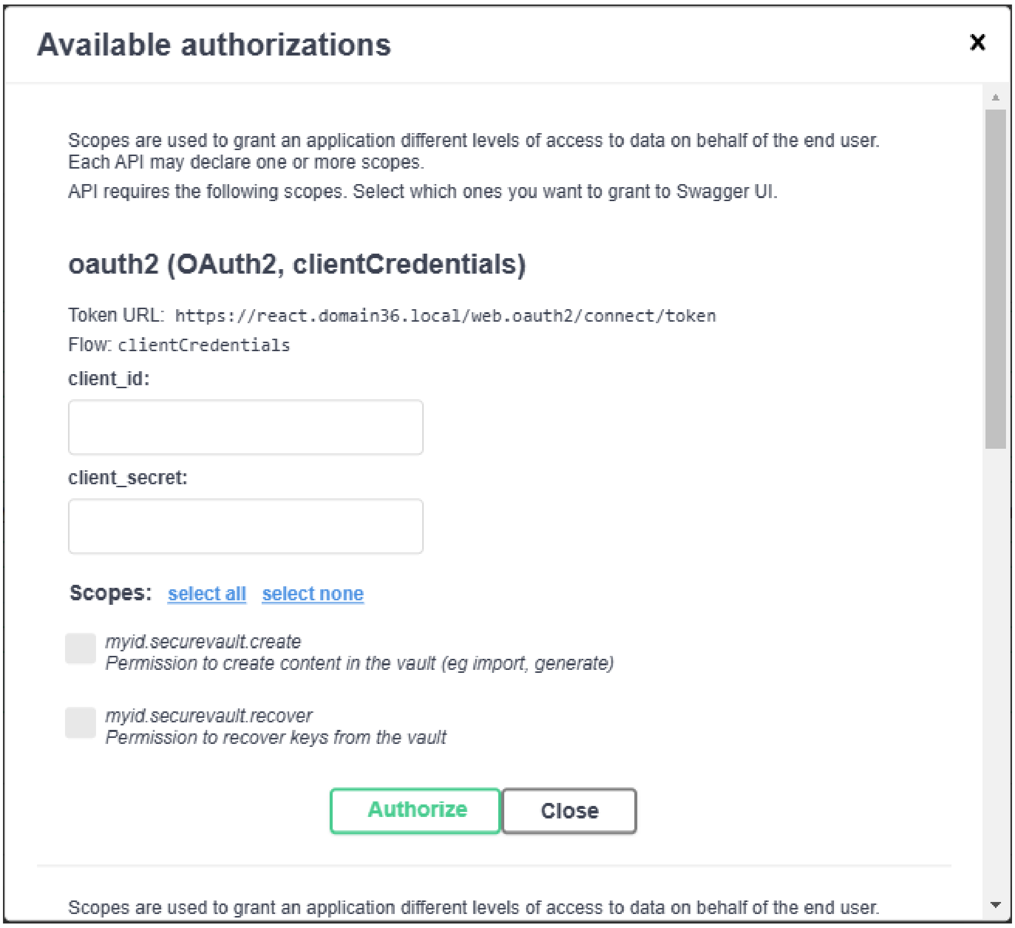
   where `<myserver>` is the name of the server hosting the MyID SecureVault web service.

2. Click the **Authorize** button.

   Note: If the **Authorize** button is not available, your administrator has prevented access by setting the `AllowTestFromDocumentation` configuration option to `false`. See section *5.1*, *Accessing the API documentation*.

3. Locate the following section:

   oauth2 (OAuth2, clientCredentials)

   Scopes are used to grant an application different levels of access to data on behalf of the end user.
   Each API may declare one or more scopes.
   API requires the following scopes. Select which ones you want to grant to Swagger UI.

   oauth2 (OAuth2, clientCredentials)

   Token URL: `https://react.domain36.local/web.oauth2/connect/token`
   Flow: `clientCredentials`

   client_id:

   client_secret:

   □ *myid.securevault.create*
   Permission to create content in the vault (eg import, generate)

   □ *myid.securevault.recover*
   Permission to recover keys from the vault

   Scopes are used to grant an application different levels of access to data on behalf of the end user.

4. Complete the following details:
   - **client_id** – type the client ID you created when configuring MyID SecureVault for OAuth2. For example:

     `myid.securevault`
   - **client_secret** – type the client secret you created when configuring MyID SecureVault for OAuth2.

     See section *3.1.1.1*, *Creating a client secret* for details.

     Note: This is the `client secret` value output by the PowerShell script, *not* the `SHA256+base64` value.
   - **Scopes** – select the scopes you need for the features you want to use.
5. Click **Authorize**.
6. Click **Close**.

You now have an access token that you can use to call the API.

5.2.1.2 Authenticating using two-way TLS

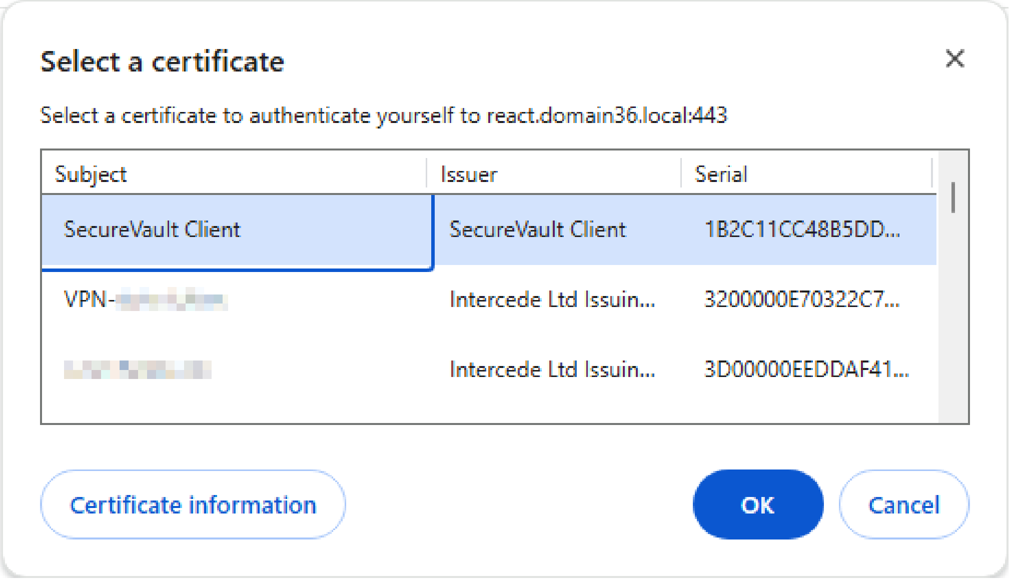
To authenticate using two-way TLS:

1. Open a web browser and navigate to the MyID SecureVault API documentation:

   `https://<myserver>/SecureVault/swagger/index.html`

   where `<myserver>` is the name of the server hosting the MyID SecureVault web service.

   Your browser prompts you to select a certificate.



Select a certificate

Select a certificate to authenticate yourself to react.domain36.local:443

Subject	Issuer	Serial
SecureVault Client	SecureVault Client	1B2C11CC48B5DD...
VPN-[redacted]	Intercede Ltd Issuin...	3200000E70322C7...
[redacted]	Intercede Ltd Issuin...	3D00000EEDDAF41...

2. Select the client certificate you configured for use with MyID SecureVault, then click **OK**.

You are now authenticated using the certificate and can call the API.

5.2.2 Authenticating to the API from PowerShell

You can call the API from a PowerShell script. You can use a script to obtain an OAuth2 access token or provide the details of the client certificate for two-way TLS.

5.2.2.1 Authenticating using OAuth2

If you are using OAuth2 authentication, the first thing you must do is obtain an access token that you can use to call the API.

The *Obtaining a server-to-server access token* in the *MyID Core API* guide in the MyID CMS documentation set provides information on obtaining access tokens.

For example, you can use the following script.

```
# Specify the domain of the server
$server = "myserver.example.com"

# Set the client secret
$secret = "072153b4-f76a-4644-abb5-e6ca0c304822"

# Set the URI of the isAlive endpoint
$uri = 'https://' + $server + '/web.oauth2/connect/token'

# Set up the header
$header = @{
    'Content-Type' = 'application/x-www-form-urlencoded'
 }

 # Set up the request
 $request  = @{
    Headers =  $header
    Uri = $uri
    Body = @{
        'grant_type' = 'client_credentials'
        'scope' = 'myid.securevault.create myid.securevault.recover'
        'client_id' = 'myid.securevault'
        'client_secret' = $secret
    }
    Method = "POST"
}

# Call the API and display the result
$result = Invoke-WebRequest @request
Write-Host $result
```

Edit the following variables before you run the script:

- `$server` – the name of your MyID web.oauth2 server.
- `$secret` – the client secret you set up on the MyID web.oauth2 server for the `myid.securevault` client.

This script obtains an access token that you can use to call the API.

5.2.2.2 Authenticating using two-way TLS

If you are using two-way TLS, you must have the client certificate installed on your PC. You can then provide the thumbnail of the client certificate when you call the API.

For example, you can use the following script to call the `IsAlive` endpoint:

```
# Specify the domain of the server
$server = "myserver.example.com"

# Set the thumbprint of the client certificate
$thumbprint = "1B28ACFA999DF9CF2915A02066644592BCC7E126"

# Set the URI of the isAlive endpoint
$uri = 'https://' + $server + '/securevault/api/Keys/isAlive'

# Set up the header
$header = @{
    'accept' = 'text/plain'
 }

 # Set up the request
 $alive  = @{
    Headers =  $header
    Uri = $uri
    Method = "POST"
    # Set the certificate thumbprint for the two-way TLS client certificate
    CertificateThumbprint = $thumbprint
}

# Call the API and display the result
$result = Invoke-WebRequest @alive
Write-Host $result
```

Edit the following variables before you run the script:

- `$server` – the name of your MyID SecureVault server.
- `$thumbprint` – the thumbprint of the client certificate.

5.3 Calling the API

When you call the API endpoints, you must use the API documentation to determine the endpoint, the headers, and the payload for the API call.

For example, MyID SecureVault provides the following endpoint to allow you to generate a new key in the MyID SecureVault keystore:

```
POST /api/Keys/generate
```

This endpoint has the following parameters that you can pass in the body:

- `alg` – the algorithm you want to use to create the key. You can find the accepted values in the Schema section of the API documentation.

  Note: RSA 1024 is listed as a supported key type in the schema; however, while you can import RSA 1024-bit keys, you cannot generate RSA 1024-bit keys.
- `label` – the label you want to use for the key.

Store this required payload as JSON and add it to the body of your request.

5.3.1 Calling the API using PowerShell and OAuth2 authentication

You can use the following PowerShell script to call the generate endpoint using OAuth2 authentication:

```
# Specify the domain of the server
$server = "myserver.example.com"

# Set the access token
$token = "<YOUR-TOKEN>"

# Set the URI of the generate endpoint
$uri = 'https://' + $server + '/SecureVault/api/Keys/generate'

# Set up the header
$header = @{
    'accept' = 'text/plain'
    'Content-Type' = 'application/json'
    'Authorization'="Bearer $token"
 }

# Set up the body of the request
$body = '{
    "alg": "RSA2048",
    "label": "My Example Key Name"
    }'

# Set up the request
$request  = @{
    Headers =  $header
    Uri = $uri
    Method = "POST"
    Body = $body
}

# Call the API and display the result
$result = Invoke-WebRequest @request
Write-Host $result
```

Edit the following variables before you run the script:

- `$server` – the name of your MyID SecureVault server.
- `$token` – the access token you have acquired from the MyID web.oauth2 server.

  Note: You do not need all the JSON returned from the web.oauth2 server (which contains additional information such as the expiry and the scope); set the `$token` variable to the value of the `"access_token"` key.

5.3.2 Calling the API using PowerShell and two-way TLS

You can use the following PowerShell script to call the generate endpoint using two-way TLS:

```
# Specify the domain of the server
$server = "myserver.domain.com"

# Set the thumbprint of the client certificate
$thumbprint = "1B28ACFA999DF9CF2915A02066644592BCC7E126"

# Set the URI of the generate endpoint
$uri = 'https://' + $server + '/SecureVault/api/Keys/generate'

# Set up the header
$header = @{
    'accept' = 'text/plain'
    'Content-Type' = 'application/json'
 }

# Set up the body of the request
$body = '{
    "alg": "RSA2048",
    "label": "My Example Key Name"
    }'

# Set up the request
$request  = @{
    Headers =  $header
    Uri = $uri
    Method = "POST"
    CertificateThumbprint = $thumbprint
    Body = $body
}

# Call the API and display the result
$result = Invoke-WebRequest @request
Write-Host $result
```

Edit the following variables before you run the script:

- `$server` – the name of your MyID SecureVault server.
- `$thumbprint` – the thumbprint of the client certificate.

5.3.3 Batch operations

MyID SecureVault provides endpoints that allow you to carry out batch operations on biometric data:

- Retrieve a set of biometric data items.

  `GET /api/Data/biometric`
- Add a batch of biometric data items to be encrypted in storage.

  `POST /api/Data/batch/biometric`
- Update a batch of biometric data items in storage.

  `PATCH /api/Data/batch/biometric`
- Delete a batch of biometric data items from storage.

  `DELETE /api/Data/biometric`

Note: Batch operations are transactional on the database; if a failure occurs, the entire batch is rolled back.

By default, the `GET`, `POST`, and `PATCH` operations are restricted to a maximum of 50 items; this is to prevent unacceptable delays, particularly when using an HSM to encrypt or decrypt the data. There is no restriction on the `DELETE` operation, as it does not use encryption or decryption.

5.3.3.1 External link IDs

For batch endpoints, all items being processed in a single call need to be for the same external link ID. If multiple external link IDs are found in the same batch operation, an error similar to the following occurs:

```
VLT10016 - External Link IDs need to be the same in the same batch.
```

External link IDs are provided by the calling system when storing data in MyID SecureVault; for example, the calling system may provide a GUID uniquely identifying a person when storing biometric data. This would mean that you could carry out batch operations on all biometric data stored for the same person, and an error would occur if you tried to carry out batch operations on biometric data for different people in the same batch.

5.3.3.2 Setting the number of records for batch operations

If you want to increase or decrease the limit on batch operations, you can edit the `"GeneralOptions":"MaxRecords"` option in the application settings file for the MyID SecureVault web service:

1. On the MyID SecureVault server, navigate to the MyID SecureVault web service folder.

   By default, this is:

   `C:\Program Files\Intercede\SecureVault\SecureVault`
2. Open the `appsettings.Production.json` file for the web service in a text editor.

   This file is the override configuration file for the `appsettings.json` file for the web service. Making your changes in this file rather than the `appsettings.json` file ensures that your configuration changes are not overwritten when you update or upgrade MyID SecureVault.

   If the `appsettings.Production.json` file does not already exist, you must create it. You can copy the existing `appsettings.json` file.
3. Edit the `MaxRecords` section to specify the maximum number of records you want to allow in a batch operation:

   ```
   "GeneralOptions": {
     "MaxRecords": 50
   },
   ```
4. Save the file.
5. Recycle the web service app pool:
   a. On the MyID SecureVault server, in Internet Information Services (IIS) Manager, select **Application Pools**.
   b. Right-click the **SecureVaultPool** application pool, then from the pop-up menu click **Recycle**.

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

6 Troubleshooting

This section contains information on troubleshooting issues you may experience when installing or using MyID SecureVault.

This section contains:

- Help with troubleshooting installation issues.

  See section *6.1*, *Installation issues*.
- Details on how to configure logging for the MyID SecureVault web service.

  See section *6.2*, *Configuring logging*.
- Details of accessing the audit report.

  See section *6.3*, *Auditing*.
- A list of error codes that may occur when using MyID SecureVault.

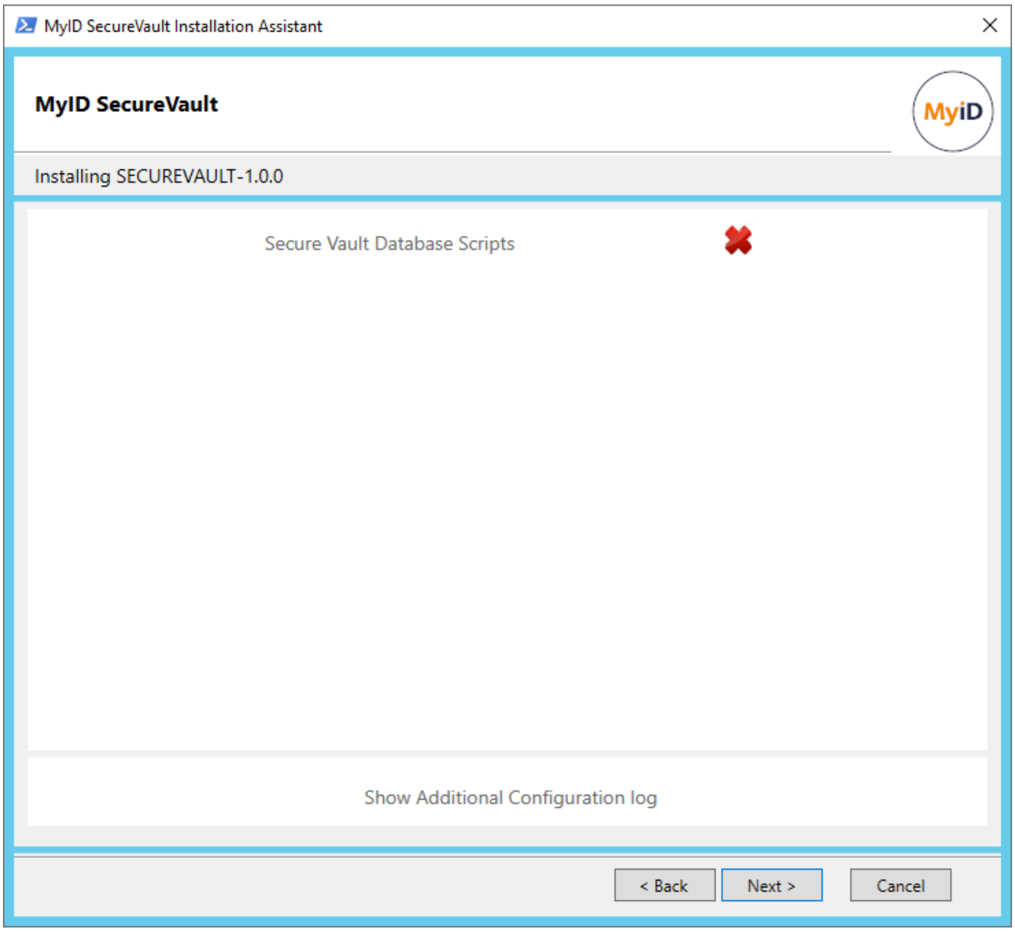
  See section *6.4*, *Error codes*.

6.1 Installation issues

You can view the installation log in the following folder:

```
<install folder>\Installer\
```

If an issue occurs during the installation, the installation stops at the stage where the issue occurred, and the installer displays an error indicator.



MyID SecureVault Installation Assistant
MyID SecureVault
MyID
Installing SECUREVAULT-1.0.0
Secure Vault Database Scripts ✖
Show Additional Configuration log
< Back
Next >
Cancel

Click **Show Additional Configuration log** to display the installer log file, which may contain information that can help you identify the issue.

If you experience installation issues, you are also recommended to run the installation from the Windows PowerShell command prompt:

1. Open a Windows PowerShell command prompt with elevated permissions.
2. Navigate to the following folder:

   ```
   <install folder>\Support Tools\SecureVaultInstallationAssistant\
   ```

3. Run the following script:

   ```
   .\SecureVaultInstallationAssistant.ps1
   ```

This provides you with some additional debug information in the console.

6.2 Configuring logging

To set up logging for the MyID SecureVault web service:

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

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

   This is the override file for the `appsettings.json` files for the MyID SecureVault web service. If this file does not already exist, you must create it in the same folder as the `appsettings.json` files.
2. Ensure that there is an entry for logging Intercede components.

   For example:

   ```
   {
       "Logging":  {
           "LogLevel":  {
               "Intercede": "Debug"
           }
       }
   }
   ```

   This must be set to at least `Debug` to allow logging. This is then further filtered by the `Log.config` file.

   Note: If you already an have `appsettings.Production.json` file, add the `Logging:LogLevel:Intercede` section to the existing file. The above example assumes that there are no other entries in the file.
3. Save the file.
4. In a text editor, open the `Log.config` file for the MyID SecureVault web service:

   ```
   C:\Program Files\Intercede\SecureVault\SecureVault\Log.config
   ```
5. Set the value of the `file` node to the output location; for example:

   ```
   <file value="C:\Logs\Intercede.MyID.SecureVault.log" />
   ```
6. 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.

7. Save the file.

Note: You must ensure that the MyID SecureVault 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.

6.3 Auditing

The `Audits` table in the MyID SecureVault audit database (by default, `SecureVaultAudit`) contains auditing details of the key operations and biometric storage operations that MyID SecureVault has carried out. The `ExternalLinkID` field in the `Audits` table specifies the identifier provided by the calling system for data items; for example. a GUID identifier for the person whose biometric data is being stored.

There is currently no user interface on this audit information, but you can use your own tools to query the MyID SecureVault audit database to return reports tailored to your own specifications.

6.4 Error codes

This section contains a list of error codes that may occur when using MyID SecureVault.

Error Code	VLT10000
Text	An error has occurred.
Details	An unexpected error has occurred. No further information is available.
Solution	Enable logging and try the operation again. The log file may contain information that helps you identify the issue. See section *6.2*, *Configuring logging*.

Error Code	VLT10001
Text	A protection error has occurred.
Details	An error has occurred when attempting to encrypt or decrypt data.
Solution	Enable logging and try the operation again. The log file may contain information that helps identify the issue. See section *6.2*, *Configuring logging*. This error may occur with a status 500 error if you have misconfigured your `appsettings.Production.json` file or if the `appsettings.Production.json` file is not accessible. If you have this error with a status of 500: • Check that your database connections settings are correct and that the database is available. • Check that the HSM settings are correct and that the HSM is available. • Check that the `appsettings.Production.json` file is correctly named, located, and configured, and that the encryption certificate thumbprint is set correctly. If you experience this error when generating a key in the ServerKeyStore: • Check that the MyID SecureVault application pool is configured to set the **Load User Profile** option to **True**; see section *2.5.1*, *Configuring the application pool*.

Error Code	VLT10002
Text	Key generation failed.
Details	MyID SecureVault failed to generate a key; this may occur both when using an HSM and when using software.
Solution	For software, check your certificate settings. See section *4.1.1*, *Generating a self-signed certificate* and section *4.1.2*, *Configuring the MyID SecureVault web service with the certificate*. For HSM key generation, check your HSM settings. See section *4.2*, *Configuring MyID SecureVault for use with an HSM*.

Error Code	VLT10003
Text	No data supplied.
Details	You have attempted to call the API but have not provided all of the information required.
Solution	Check the parameters required for the API endpoint in the API documentation; see section *5.1*, *Accessing the API documentation*.

Error Code	VLT10004
Text	Key algorithm is invalid.
Details	You have attempted to call the API but have specified a key algorithm that is not recognized or is not supported. You cannot generate RSA 1024 keys.
Solution	Check that the key algorithm you want to use is supported, then try again.

Error Code	VLT10005
Text	Invalid data supplied.
Details	You have attempted to call the API but the data your provided was invalid.
Solution	Check the data you are passing. In particular, check whether you are passing hashed data or if MyID SecureVault is hashing the data.

Error Code	VLT10006
Text	A database error occurred.
Details	A problem has occurred with the database.
Solution	Check that the database is running, and that there are no connection issues.

Error Code	VLT10007
Text	Item not found.
Details	You have attempted to call the API, but have specified an item (for example, the ID of a key) that MyID SecureVault cannot find in the database.
Solution	Check the IDs that you are passing to the API.

Error Code	VLT10008
Text	An error occurred parsing or processing the PKCS12.
Details	MyID SecureVault has experienced a problem working with the provided PKCS #12 PFX file.
Solution	Check that the PKCS #12 PFX file uses supported encryption and hashing algorithms. Supported encryption algorithms: • AES128 • AES192 • AES256 • 3DES Supported hashing algorithms: • SHA1 • SHA256

Error Code	VLT10009
Text	Unable to encrypt key.
Details	MyID SecureVault was unable to encrypt the key for transport.
Solution	Enable logging and try the operation again. The log file may contain information that helps you identify the issue. See section *6.2*, *Configuring logging*.

Error Code	VLT10010
Text	Unable to decrypt key.
Details	MyID SecureVault was unable to decrypt the key stored in the database.
Solution	Enable logging and try the operation again. The log file may contain information that helps you identify the issue. See section *6.2*, *Configuring logging*.

Error Code	VLT10011
Text	Error communicating with HSM.
Details	MyID SecureVault was unable to communicate with the HSM.
Solution	Check your HSM configuration; see section *4.2*, *Configuring MyID SecureVault for use with an HSM*.

Error Code	VLT10012
Text	An error occurred generating the PKCS12.
Details	MyID SecureVault was unable to generate the requested PKCS #12 file.
Solution	Enable logging and try the operation again. The log file may contain information that helps you identify the issue. See section *6.2*, *Configuring logging*.

Error Code	VLT10013
Text	Data Type not allowed.
Details	A data type has been supplied that is not in the allowed list in the application settings file.
Solution	Supply a valid data type. Currently, only `biometric` is supported as a data type.

Error Code	VLT10014
Text	No Items in request.
Details	No items have been supplied into a batch endpoint, so there is nothing to process.
Solution	Supply some valid items.

Error Code	VLT10015
Text	Invalid ID.
Details	The ID of the record specified is invalid.
Solution	Supply a valid ID, which is a GUID.

Error Code	VLT10016
Text	External Link IDs need to be the same in the same batch.
Details	For Batch endpoints, all items being processed in a single call need to be for the same external link ID. Multiple external link IDs have been found in the same batch operation.
Solution	Check the external link IDs of the supplied items. See section *5.3.3.1*, *External link IDs* for more information.

Error Code	VLT10017
Text	There are more records than the maximum allowed.
Details	There are more than the maximum number of items in the batch, either supplied in the Add or Update operation, or returned from the GET operation.
Solution	If it is a GET operation, refine your criteria; if its an Add or Update, provide the correct number of items for each batch. The default maximum is 50 items. You can configure the maximum number of items to include in a batch using the `MaxRecords` option in the application settings file; see section *5.3.3.2*, *Setting the number of records for batch operations*.

7 Changing passwords

Your organization may require that you change your passwords regularly. To assist you with this process, you can use the Password Change Tool. This utility was designed for MyID CMS, but also allows you to change the MyID SecureVault web service user password.

The utility is provided in the following folder:

```
<install folder>\Support Tools\SecureVaultInstallationAssistant\
utilities\Password Change Tool\
```

See the readme provided with the utility for installation instructions.

By default, the PCT log file is saved in the same directory as the executable; however, you can update the `Log.config` file in the tool folder to give it a different file path.

To change the log file path, edit the value of the file parameter:

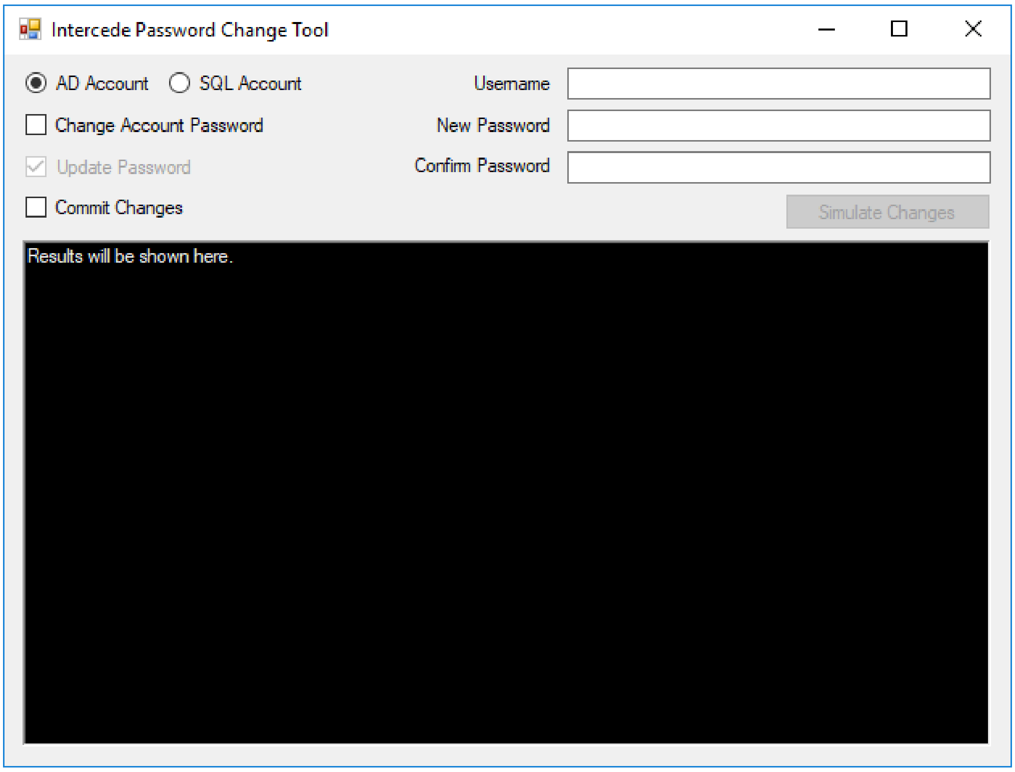
```
<file value="PasswordChangeTool.log" />
```

Note: If you need to change the SQL Authentication password for the MyID SecureVault databases, you must change the password in SQL Server Management Studio, then encrypt and store the password in the MyID SecureVault application settings file; see section 2.7, *Configuring the database connection manually*. You cannot use the Password Change Tool for this purpose.

7.1 Changing the MyID SecureVault web service user account password

To change the MyID SecureVault web service user account password:

1. In Windows Explorer, navigate to the folder where you have installed the Password Change Tool.
2. Hold the Shift key, right-click the `PasswordChangeTool.exe` file, then select **Run as different user**.
3. Enter domain administrator credentials to start the tool.
4. Ensure **AD Account** is selected.



Intercede Password Change Tool

● AD Account ○ SQL Account

□ Change Account Password

☑ Update Password

□ Commit Changes

Username

New Password

Confirm Password

```
Results will be shown here.
```

5. If you want to change the password on the Active Directory at the same time, select the **Change Account Password** option.

   Note: The user under which you are running the PCT must have the appropriate permissions for this to be successful.

   If you have already changed the password on the Active Directory, and just want to update the MyID SecureVault web service to use the new password, make sure that **Change Account Password** is *not* selected.
6. Type the **Username** and the **New Password** for the user, then type the password again in the **Confirm Password** field.

   For usernames, you can enter either the fully-qualified user name of the user in either the format `domain\user`, or `user@domain` – if you enter username on its own without specifying the domain, the domain of the currently logged on user is used.
7. Ensure the **Commit Changes** option is not selected.

8. Click **Simulate Changes**.

   The PCT checks the user details with the Active Directory service. The PCT abandons the change at this point if either the **Username** cannot be found or the specified **Password** is incorrect. In any of these cases, the PCT makes no changes to your system.

9. Verify that no errors (shown in red) are displayed. All the actions that would have been performed are shown in green.

   Note: Confirm that all of the proposed password changes are acceptable before continuing.

10. Select the **Commit Changes** option.

    This changes the **Simulate Changes** button to **Commit Changes**.

    Note: Log out of MyID on all clients before proceeding.

11. Click **Commit Changes**.

    The PCT makes all the changes to your system. The MyID SecureVault IIS application pool is recycled.

12. Verify that there are no errors shown.

Appendix A Configuring MyID CMS for MyID SecureVault

You can integrate MyID SecureVault with MyID CMS, which allows MyID CMS to use MyID SecureVault to generate, store, and recover keys, or to store and retrieve biometric data.

This chapter contains the following:

- Requirements for integrating MyID CMS with MyID SecureVault.

  See section *A.1*, *Prerequisites for integration between MyID CMS and MyID SecureVault*.
- Details of setting up the external systems in MyID CMS that provide the links to up to ten instances of MyID SecureVault.

  See section *A.2*, *Setting up the MyID SecureVault external system*.
- Information on setting up certificate policies within MyID CMS.

  See section *A.3*, *Setting up certificate policies to use MyID SecureVault*.
- Instructions for importing certificates into MyID and storing their keys in MyID SecureVault.

  See section *A.4*, *Importing keys into MyID SecureVault through MyID CMS*.
- Instructions for recovering keys from MyID SecureVault.

  See section *A.5*, *Recovering keys from MyID SecureVault through MyID CMS*.
- Details of MyID SecureVault reports available in the MyID Operator Client.

  See section *A.6*, *SecureVault reports*.
- Troubleshooting for connection issues.

  See section *A.7*, *Connection issues*.

Further configuration is required if you want to use MyID SecureVault to store biometric data; see section *Appendix C*, *Configuring MyID CMS for biometric storage* for details.

A.1 Prerequisites for integration between MyID CMS and MyID SecureVault

To set up your environment, you require the following:

- An installation of MyID CMS 12.13 or later.
- An installation of MyID SecureVault.
- A certificate authority that you can use to issue certificates.
- Optionally, an HSM.

Note: Some features require specific versions of MyID CMS, or require additional updates for MyID CMS.

Feature	MyID CMS version
Integration with MyID CMS	MyID CMS 12.14 or later.
Storage and recovery of ECC keys	MyID CMS 12.14 or later.
MyID for Key Escrow	MyID CMS 12.14 or later.
Multiple instances of MyID SecureVault	MyID CMS 12.16 or later.
Storage of biometric data	MyID CMS 12.17 or later (currently scheduled).

MyID CMS 12.14 supports the following feature with an additional update:

- Multiple instances of MyID SecureVault.

  Requires `UPDATE-12.14.0.1`.

MyID CMS 12.13 supports the following features with additional updates:

- Integration with MyID CMS.

  Requires `UPDATE-12.13.0.1`.
- Multiple instances of MyID SecureVault.

  Requires `UPDATE-12.13.0.2`.
- Storage of biometric data.

  Requires `UPDATE-12.13.0.2`.

A.2 Setting up the MyID SecureVault external system

Before you set up the external system with MyID CMS to integrate with MyID SecureVault, you must decide which method of authentication to use (OAuth2 or two-way TLS) and configure your servers appropriately; see section *3*, *Setting up authentication for the MyID SecureVault web service* for details.

To set up the external system:

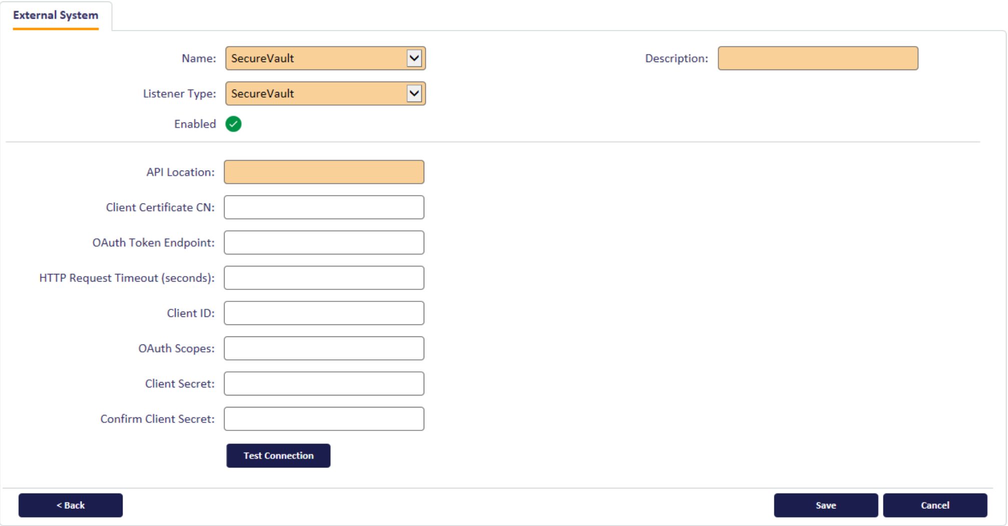
1. In MyID Desktop, from the **Configuration** category, select **External Systems**.

   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.

2. Click **New**.

3. From the **Listener Type** drop-down list, select **SecureVault**.

   The MyID SecureVault options appear.



Name: SecureVault

Listener Type: SecureVault

Enabled ✓

4. Select a **Name** from the list.

   You can create up to ten external systems with a **Listener Type** of **SecureVault**. The names are fixed as `SecureVault` and `SecureVault 2` through `SecureVault 10`. You can specify different SecureVault servers for different certificate policies; for example, if you have three certificate policies, you could archive the encryption certificate in SecureVault, the authentication certificate in SecureVault 2, and leave the signing certificate unarchived. You can also use a separate instance of MyID SecureVault to store biometric data.

   Take a note of the number of the instance you are using; you need this information when you configure your certificate policies to determine which instance of MyID SecureVault you want to use to archive the keys for that policy, or when setting the **Biometric storage location** option to configure MyID CMS to store its biometric data in MyID SecureVault.

   Note: Support for multiple instances of MyID SecureVault is available with MyID CMS 12.16 or later. Earlier versions of MyID CMS supported only a single instance of

SecureVault with a fixed name; however, updates are available for MyID CMS 12.13 and 12.14 to add support for multiple instances of MyID SecureVault. See section *A.1*, *Prerequisites for integration between MyID CMS and MyID SecureVault* for details.

5. Type a **Description** for the external system.
6. Provide the **API Location**.

   This is the address of the MyID SecureVault web service; for example:

   `https://myserver.example.com/securevault`
7. Optionally, set the **HTTP Request Timeout (seconds)**.

   If you leave this option blank, the default is 30 seconds.
8. Configure your authentication settings.
   - If you are using two-way TLS, set the following:
     - **Client Certificate CN** – type the CN of the client certificate you have installed on your MyID application server.

       For example:

       `SecureVault Client`

       Note: Do not include the `CN=` part included in the PowerShell script output.

     See section *3.2*, *Setting up two-way TLS authentication* for details of setting up two-way TLS.
   - If you are using OAuth2 authentication, set the following:
     - **OAuth Token Endpoint** – type the location of the token endpoint for your OAuth2 server.

       For example:

       `https://myserver.example.com/web.oauth2/connect/token`
     - **Client ID** – type the ID you created for the MyID SecureVault client.

       For example:

       `myid.securevault`
     - **OAuth Scopes** – type the scopes required for access to the MyID SecureVault server features, separated by spaces.

       For example:

       `myid.securevault.create myid.securevault.recover`
     - **Client Secret** – type and confirm the client secret you created for the MyID SecureVault web service.

       See section *3.1.1.1*, *Creating a client secret* for details.

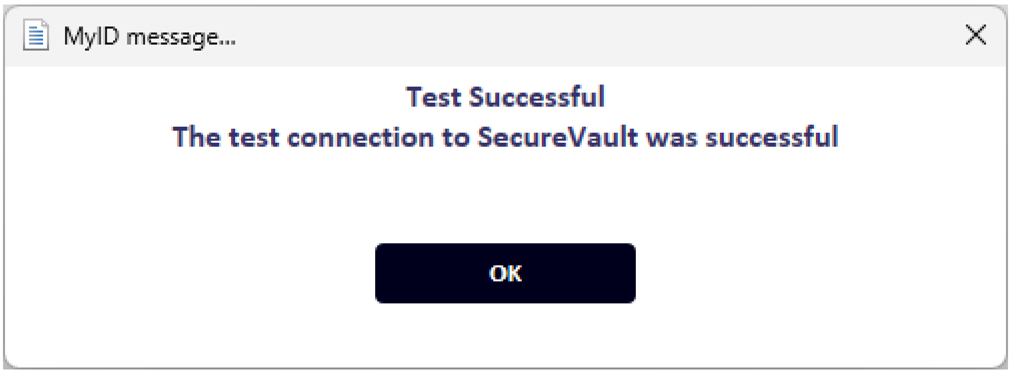
       Note: This is the `client secret` value output by the PowerShell script, *not* the `SHA256+base64` value.

       If you edit your external system, you must re-enter and confirm your client secret, even if it has not changed.

     See section *3.1*, *Setting up OAuth2 authentication* for details of setting up OAuth2 authentication.

9. Click **Test Connection**.

   If MyID Desktop can successfully communicate with the MyID SecureVault web service, it displays a confirmation dialog.

   Test Successful
   The test connection to SecureVault was successful

   If the test connection fails, you can view more details about the error. See section *A.7*, *Connection issues* for more help.

   Click **OK** to close the confirmation dialog.

10. Click **Save**.

A.3 Setting up certificate policies to use MyID SecureVault

Note: MyID SecureVault currently supports the generation, storage, and recovery of RSA 2048, 3072, and 4096 bit keys, and ECC P256, P384, and P521 curves. You can import and recover (but not generate) RSA 1024 bit keys.

To set up a certificate policy to use MyID SecureVault for archiving:

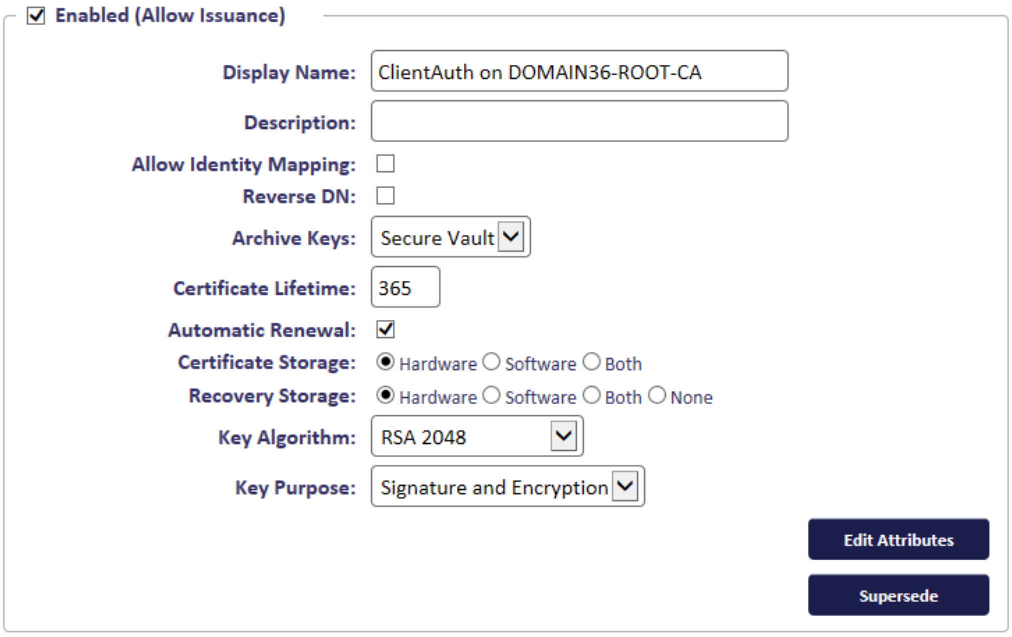
1. In MyID Desktop, from the Configuration category, select the **Certificate Authorities** workflow.

   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. From the **CA Name** list, select the certificate authority that contains the certificate policy you want to work with.
3. From the list of **Available Certificates**, select the certificate policy you want to work with.
4. From the **Archive Keys** drop-down list, select the appropriate SecureVault server.

   You can use different SecureVault servers for different certificate policies; for example, if you have three certificate policies, you could archive the encryption certificate in **Secure Vault**, the authentication certificate in **Secure Vault 2**, and leave the signing certificate unarchived.

   The **Certificate Authorities** workflow lists all ten possible instances of MyID SecureVault, whether or not you have configured them in the **External Systems** workflow; make sure you use the correct instance to match the instance you have already configured.

   Note: Support for multiple instances of MyID SecureVault is available with MyID CMS 12.16 or later. Earlier versions of MyID CMS supported only a single instance of SecureVault with a fixed name; however, updates are available for MyID CMS 12.13 and 12.14 to add support for multiple instances of MyID SecureVault.



☑ Enabled (Allow Issuance)

Display Name: ClientAuth on DOMAIN36-ROOT-CA

Description:

Allow Identity Mapping: □

Reverse DN: □

Archive Keys: Secure Vault

Certificate Lifetime: 365

Automatic Renewal: ☑

Certificate Storage: ● Hardware ○ Software ○ Both

Recovery Storage: ● Hardware ○ Software ○ Both ○ None

Key Algorithm: RSA 2048

Key Purpose: Signature and Encryption

Note: You can issue soft certificates using ECC keys only if you have configured the certificate policy to archive the keys in MyID SecureVault; issuing soft certificates using ECC keys is not otherwise supported in MyID CMS.

5. Click **Save**.

A.4 Importing keys into MyID SecureVault through MyID CMS

You can use the MyID Core API to import certificates that were issued by a different system into MyID CMS; this allows MyID CMS to manage the certificates as if they had been issued by MyID. This feature is available only through the API, not through the MyID Operator Client.

You can use this feature to migrate keys from an HSM to MyID SecureVault; if you export the key from your HSM as a PKCS #12 PFX file (which contains the private key and corresponding certificate), you can then use the MyID Core API to import this file into MyID CMS.

If you configure the certificate policy in the **Certificate Authorities** workflow in MyID CMS to specify MyID SecureVault as the **Archive Keys** mechanism, MyID CMS uses MyID SecureVault to provide secure storage and recovery of the keys for the certificate.

Note: You can import ECC keys into MyID CMS only if you have configured the certificate policy to use MyID SecureVault to archive the keys; importing ECC keys into MyID CMS is not otherwise supported.

For information on importing certificates using the MyID Core API, see the *Importing certificates* chapter of the *MyID Core API* guide in the MyID CMS documentation set.

Note: You can also import keys through the MyID SecureVault API; see section *5*, *Using the MyID SecureVault API*.

A.5 Recovering keys from MyID SecureVault through MyID CMS

If you have imported or issued a certificate through MyID CMS using a certificate policy that you have configured to use MyID SecureVault, you can subsequently recover the keys for that certificate from the MyID SecureVault key store.

Once you have archived they key in MyID SecureVault, any operation in MyID CMS that needs to recover that key (for example, administrator key recovery, collecting smart cards, and so on) recovers it from the MyID SecureVault key store.

Note: You can also recover keys through the MyID SecureVault API; see section *5*, *Using the MyID SecureVault API*.

A.6 SecureVault reports

The following reports are available in the MyID Operator Client:

- SecureVault Usage.

  See section *A.6.1*, *SecureVault Usage report*.
- SecureVault Biometric Usage.

  See section *A.6.2*, *SecureVault Biometric Usage report*.

A.6.1 SecureVault Usage report

This report is available in the Reports category in the MyID Operator Client, and returns a count of the number of keys archived in each instance of MyID SecureVault.

This report is available only if you have been granted access. See the *Granting access to reports* section in the *MyID Operator Client* guide for details.

A.6.1.1 Search criteria

There are no search criteria for this report.

A.6.1.2 Report fields

The report contains the following fields:

Field	Description
Vault Name	The name of the MyID SecureVault instance. You can configure up to ten instances of MyID SecureVault for integration with MyID CMS.
Keys	The number of keys archived in the instance of MyID SecureVault.

A.6.1.3 Running the report through the API

The SecureVault Usage report has the report ID `290037`.

See the *Running reports through the MyID Core API* section in the *MyID Operator Client* guide for more details of running reports through the MyID Core API.

A.6.2 SecureVault Biometric Usage report

This report is available in the Reports category in the MyID Operator Client, and returns a count of the number of biometric samples stored in MyID SecureVault.

Note: This report does not query the MyID SecureVault database directly; it returns the number of biometric samples in the MyID CMS database that are marked as stored in MyID SecureVault.

This report is available only if you have been granted access. See the *Granting access to reports* section in the *MyID Operator Client* guide for details.

A.6.2.1 Search criteria

There are no search criteria for this report.

A.6.2.2 Report fields

The report contains the following fields:

Field	Description
External Archive Reference	The ID of the MyID SecureVault system in which the biometric records are stored.
Count	The number of biometric samples stored in the instance of MyID SecureVault.

A.6.2.3 Running the report through the API

The SecureVault Usage report has the report ID `290038`.

See the *Running reports through the MyID Core API* section in the *MyID Operator Client* guide for more details of running reports through the MyID Core API.

A.7 Connection issues

You may experience the following errors when attempting to connect to the MyID SecureVault web service through the MyID CMS **External Systems** workflow.

- If you have specified the wrong **Client Certificate CN**:

  ```
  Error connecting to SecureVault server - Error: 0x80072f0c Info: Failed
  to Send http(s) to
  https://react.domain36.local/securevault/api/Keys/isAlive
  Exception raised in function:
  SendHttpRequest::SendHTTPRequest In file SendHttpRequest.cpp at line 97
  (std)
  ```

  Make sure the **Client Certificate CN** is correct. Do not include the `CN=` component.

- If the client certificate is not installed on the MyID SecureVault server:

  ```
  Error connecting to SecureVault server - Info: POST to:
  https://react.domain36.local/securevault/api/Keys/isAlive failed,
  response: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html
  xmlns="http://www.w3.org/1999/xhtml"> <head> <title>IIS 10.0 Detailed
  Error - 403.16 - Forbidden</title>
  ```

  Make sure the client certificate is installed to the **Local Computer\Trusted Root Certification Authorities** store on the MyID SecureVault server.

- If you have specified an incorrect **Client Secret** or **Client ID**:

  ```
  Error connecting to SecureVault server - Info: POST to:
  https://react.domain36.local/web.oauth2/connect/token failed, response:
  {"error":"invalid_client"} Exception raised in function:
  SendHttpRequest::SendHTTPRequest in file SendHttpRequest.cpp at line
  116 (std)
  ```

  Make sure the client secret and client ID are correct. If you are editing the external system, you must re-enter the client secret, even if it has not changed.

- If you have specified the incorrect **OAuth Scopes**:

  ```
  Error connecting to SecureVault server - Info: POST to:
  https://react.domain36.local/web.oauth2/connect/token failed, response:
  {"error":"invalid_scope"} Exception raised in function:
  SendHttpRequest::SendHTTPRequest in file SendHttpRequest.cpp at line
  116 (std)
  ```

  The scopes must be:

  ```
  myid.securevault.create myid.securevault.recover
  ```

  Use a space to separate the scope names.

- If you have specified the wrong **OAuth Token Endpoint**:

  ```
  Error connecting to SecureVault server - Info: POST to:
  https://react.domain36.local/web.oauth3/connect/token failed, response:
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html
  xmlns="http://www.w3.org/1999/xhtml"> <head> <title>IIS 10.0 Detailed
  Error - 404.0 - Not Found</title>
  ```

  Make sure you have specified the whole token endpoint, not just the server name. For example:

  ```
  https://myserver.example.com/web.oauth2/connect/token
  ```

- If you have specified the wrong **API Location**:

  ```
  Error connecting to SecureVault server - Info: POST to:
  https://react.domain36.local/securevaults/api/Keys/isAlive failed,
  response: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html
  xmlns="http://www.w3.org/1999/xhtml"> <head> <title>IIS 10.0 Detailed
  Error - 404.0 - Not Found</title>
  ```

  Make sure you have specified the correct URL for the MyID SecureVault web service; for example:

  ```
  https://myserver.example.com/securevault
  ```

Appendix B Configuring MyID CMS for Key Escrow

You can integrate MyID SecureVault with MyID CMS to provide key archival for certificates you issue through your MyID CMS system.

MyID CMS provides key escrow operations that you are recommended to restrict to a limited number of operators, given that they can recover keys for any user within their scope. For additional separation of responsibilities, you may want to set up a completely separate MyID CMS system to carry out these key recovery operations, and prevent them from being carried out on your production MyID CMS system.

If you have multiple SecureVault instances, you can use different MyID CMS systems for key recovery, or use the same MyID CMS system for key recovery for all of your SecureVault instances.

This chapter describes setting up an additional MyID CMS system for key recovery, and contains the following:

- An overview of the MyID CMS for Key Escrow system.

  See section *B.1*, *Overview*.
- Prerequisites for using a MyID CMS for Key Escrow system.

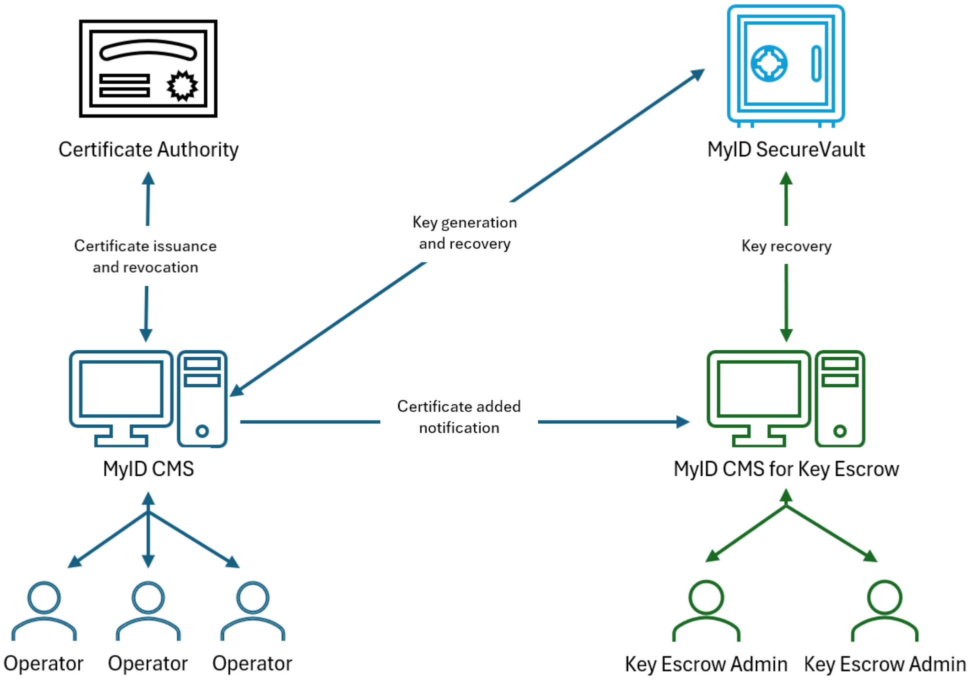
  See section *B.2*, *Prerequisites for key escrow*.
- Setting up the MyID CMS for Key Escrow system, including setting up OAuth2 authentication, setting up a connection to MyID SecureVault, configuring your imported certificate policy, setting up a key recovery credential profile, and configuring your operator permissions.

  You must configure your MyID CMS for Key Escrow system before you can configure your production MyID CMS system.

  See section *B.3*, *Configuring the MyID CMS for Key Escrow system*.
- Setting up the production MyID CMS system, including setting up the notification to communicate with the MyID CMS for Key Escrow system, setting up a connection to MyID SecureVault, setting up your certificate policies to archive keys in MyID SecureVault and generate notifications to the MyID CMS for Key Escrow system, and restricting access to key recovery workflows.

  See section *B.4*, *Configuring the production MyID CMS system*.

B.1 Overview



The MyID CMS for Key Escrow system incorporates the following:

- The MyID CMS production system.

  This is the primary MyID CMS system that your operators use for day-to-day credential lifecycle operations; however, you do not want to allow access to operations that allow you to collect a key recovery job for any user. Instead, you would prefer to silo these operations in a separate instance of MyID CMS used exclusively for key recovery.

  Whenever you add a certificate to this system, either by importing it or issuing a new certificate from the CA, the private key is stored securely in the SecureVault system. The MyID CMS production system then sends a notification to the MyID CMS for Key Escrow system containing details of that certificate (but not its private keys). The MyID CMS for Key Escrow system uses the information from the notification to add a record for the certificate to its own database, along with a minimal user record (if it does not already exist) for the certificate owner, based on the information from the certificate. This notification also contains the reference for the stored key in the SecureVault system, so that the MyID CMS for Key Escrow system can recover the key from SecureVault if required.

- The MyID CMS for Key Escrow system.

  This is an additional MyID CMS installation that is used only for collecting key recoveries. Because the MyID CMS for Key Escrow system has been notified of every certificate that has been added to the MyID CMS production system, a key escrow administrator with access to this system can request the recovery of the private keys from the MyID SecureVault system.

  Note: Because you can have multiple SecureVault servers, you may potentially have multiple MyID CMS for Key Escrow systems; when you configure the notifications from the MyID CMS production system to the MyID CMS for Key Escrow system, you must make sure that you send the notification to the appropriate instance.
- MyID SecureVault.

  MyID SecureVault provides a secure key archival module that allows you to store, generate, and recover private keys. It accepts connections from both the MyID CMS production system (for key generation, key storage, and key recovery) and the MyID CMS for Key Escrow system (for key recovery).
- The certificate authority.

  The production MyID CMS system requests certificates from the CA and carries out credential lifecycle operations such as suspension and revocation.

This configuration limits escrow key recoveries to administrators on the MyID CMS for Key Escrow system.

B.2 Prerequisites for key escrow

To set up your environment, you require the following:

- An installation of MyID CMS 12.14 or later.

  This is your MyID CMS production system.
- An installation of MyID SecureVault.
- A certificate authority that you can use to issue certificates.
- A second installation of MyID CMS 12.14 or later.

  This is your MyID CMS for Key Escrow system.
- Optionally, an HSM.

Note: If you want to connect MyID CMS to more than one MyID SecureVault system, you require MyID CMS 12.16 or later. Earlier versions of MyID CMS supported only a single instance of SecureVault with a fixed name; however, updates are available for MyID CMS 12.13 and 12.14 to add support for multiple instances of MyID SecureVault.

If you want to implement this system on a MyID CMS 12.13 system, you must install the following updates on both the MyID CMS production system and the MyID CMS for Key Escrow system.

- `UPDATE-12.13.0.1` – this update adds support for the REST Certificate Added notification, extends the **Certificate Authorities** workflow to allow you to specify the notification to send to the MyID CMS for Key Escrow system, and extends the MyID Core API to allow you to pass a reference to the archived key in MyID SecureVault when importing a certificate.

  Note: The updates from `UPDATE-12.13.0.1` are incorporated into MyID CMS 12.14 and later.
- `UPDATE-12.13.0.2` – this update adds support for multiple instances of MyID SecureVault as well as support for storing biometric data in MyID SecureVault.

B.3 Configuring the MyID CMS for Key Escrow system

The MyID CMS for Key Escrow system is an additional installation of MyID CMS that you use for key recovery operations only. You must configure this system before you can configure your production MyID CMS system to communicate with it.

B.3.1 Setting up authentication

To configure your MyID CMS for Key Escrow system to receive notifications, you must set it up for server-to-server OAuth2 authentication; see the *Server-to-server authentication* section in the *MyID Core API* guide.

Make sure you grant the user account permissions to the `/api/Certificates/import` endpoint; see the *Accessing the API* features section in the *MyID Core API* guide.

When you set up server-to-server authentication, take a note of the following details, which you need to configure the production MyID CMS system to communicate with the MyID CMS for Key Escrow system:

- **API Location** – the base URL of the MyID Core API on the MyID CMS for Key Escrow system.

  For example:

  `https://escrow.example.com/rest.core`
- **OAuth Token Endpoint** – the URL of the token endpoint on the MyID web.oauth2 web service on the MyID CMS for Key Escrow system.

  For example:

  `https://escrow.example.com/web.oauth2/connect/token`
- **Client ID** – the client ID that you create when you set up server-to-server authentication on the MyID CMS for Key Escrow system.

  For example:

  `myid.mysystem`
- **Requested Scopes** – the scope you configure when you set up server-to-server authentication.

  Typically, this is:

  `myid.rest.basic`
- **Client Secret** – the client secret you create when you set up server-to-server authentication.

  Note: This is the plain text client secret, not the Base64-encoded SHA-256 hash.

B.3.2 Setting up a connection to MyID SecureVault

You must use the **External Systems** workflow to set up a connection to your MyID SecureVault system.

See section *A.2*, *Setting up the MyID SecureVault external system* for details.

B.3.3 Setting up your certificate policy

You must configure a certificate policy that the MyID CMS for Key Escrow system uses to store the imported certificates from the production MyID CMS system.

Because the MyID CMS for Key Escrow system does not have access to the certificate authority, you must use the Unmanaged certificate authority within the **Certificate Authorities** workflow. The Unmanaged Imported profile is provided for this purpose.

To set up the certificate policy:

1. In MyID Desktop, from the **Configuration** category, select **Certificate Authorities**.

   You can also launch this workflow from the MyID Operator Client; from the **More** category, select **Connections and Notifications > Certificate Authorities**.
2. From the **CA Name** drop-down list, select **Unmanaged**.
3. Click **Edit**.
4. In the **Available Certificates** list, select **Unmanaged Imported**.

   Note: This profile is provided for imported unmanaged certificates; if you need more than one certificate profile, contact customer support quoting reference SUP-229 for more information on extending the number of unmanaged policies available.
5. Select the **Enabled** option.
6. From the **Archive Keys** drop-down list, select the appropriate SecureVault server.

   You can use different SecureVault servers for different certificate policies; for example, if you have three certificate policies, you could archive the encryption certificate in **Secure Vault**, the authentication certificate in **Secure Vault 2**, and leave the signing certificate unarchived.

   The **Certificate Authorities** workflow lists all ten possible instances of MyID SecureVault, whether or not you have configured them in the **External Systems** workflow; make sure you use the correct instance to match the instance you have already configured.

   Note: Support for multiple instances of MyID SecureVault is available with MyID CMS 12.16 or later. Earlier versions of MyID CMS supported only a single instance of SecureVault with a fixed name; however, updates are available for MyID CMS 12.13 and 12.14 to add support for multiple instances of MyID SecureVault.
7. Configure the rest of the settings for the certificate profile to match the certificates you are going to import from the production MyID CMS system.

   You can also change the **Name** and **Description** of the certificate policy, if you want.
8. Click **Save**.

You must find the object ID of the certificate policy that you are using so that you can configure the production MyID CMS system with its details. If you are using the Unmanaged Imported policy, you can run the following SQL against the MyID CMS for Key Escrow database:

```
select ObjectID from CertPolicies where Template='Unmanaged Imported'
```

B.3.4 Setting up a key recovery credential profile

You must set up a credential profile for key recovery. See the *Setting up the credential profile for key recovery* section in the *Administration Guide* for details.

B.3.5 Setting up operator permissions

You must create an operator account that has a role with permissions to access the key recovery workflows, and has sufficient scope to recover keys for any users who require key recovery.

See the *Key recovery* section in the *Administration Guide* for details of the necessary workflows.

You can configure multi-party authorization of key recovery requests by configuring your key recovery credential profile to require validation and setting up additional operator accounts that can validate key recovery operations.

B.4 Configuring the production MyID CMS system

Once you have configured the MyID CMS for Key Escrow system, you can configure your production MyID CMS system to send notifications to the key escrow system whenever a certificate is added to the production system.

B.4.1 Setting up the certificate notification

To set up the certificate notification, you require the OAuth2 details you configured on the MyID CMS for Key Escrow system; see section *B.3.1*, *Setting up authentication*.

Note: Because you can have multiple SecureVault servers, you may potentially have multiple MyID CMS for Key Escrow systems; when you configure the notifications from the MyID CMS production system to the MyID CMS for Key Escrow system, you must make sure that you send the notification to the appropriate instance.

To create the certificate external system:

1. 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.
2. Click **New**.
3. From the **Listener Type** drop-down list, select **RESTService**.
4. Complete the following details:
   - **Name** – type a name for your REST notification external system.

     You can create multiple notifications if you want to send the notifications for different certificate policies to different MyID Key for Escrow systems; provide a descriptive name that you can specify in the certificate policy settings.

     Note: For MyID 12.13 systems, the name must be:

     `REST Certificate Added`

     The name is case sensitive and you can use only one notification.
   - **Description** – type a description for the external system.
   - **Enabled** – select this option to enable the notification.
   - **Mapping File** – select the following mapping file:

     `RESTCertificateAddedMyID`

     Note: It is important that you select the `RESTCertificateAddedMyID.xml` mapping file, which is configured for sending notifications to another MyID CMS system; in this case, your MyID CMS for Key Escrow system. The `RESTCertificateAdded.xml` mapping file contains a generic endpoint and data mapping, and is not suitable for MyID CMS for Key Escrow.
   - **Notification** – select **REST Certificate Added** from the drop-down list.

     Note: For MyID 12.13 systems, this setting does not exist. The notification is determined from the **Name** you provide.

- **API Location** – type the base URL of the rest.core web service on the MyID CMS for Key Escrow server.

  For example:

  ```
  https://escrow.example.com/rest.core
  ```

- **OAuth Token Endpoint** – The URL of the token endpoint on the MyID web.oauth2 web service on the MyID CMS for Key Escrow system.

  For example:

  ```
  https://escrow.example.com/web.oauth2/connect/token
  ```

- **Client ID** – the client ID that you created when you set up server-to-server authentication on the MyID CMS for Key Escrow system.

  For example:

  ```
  myid.mysystem
  ```

- **Requested Scopes** – the scope you configured when you set up server-to-server authentication.

  Typically, this is:

  ```
  myid.rest.basic
  ```

- **Client Secret** – the client secret you created when you set up server-to-server authentication.

  Note: This is the plain text client secret, not the Base64-encoded SHA-256 hash.

  If you are using a client secret, you do not need to provide a **Bearer token**.

5. Click **Save**.

B.4.2 Setting up a connection to MyID SecureVault

You must use the **External Systems** workflow to set up a connection to your MyID SecureVault system.

See section *A.2*, *Setting up the MyID SecureVault external system* for details.

B.4.3 Setting up your certificate policies

You must configure your certificate policies to archive their keys in MyID SecureVault, and specify the notification to use when a certificate using this policy is added to the system.

To set up a certificate policy:

1. In MyID Desktop, from the **Configuration** category, select **Certificate Authorities**.

   You can also launch this workflow from the MyID Operator Client; from the **More** category, select **Connections and Notifications > Certificate Authorities**.
2. Select the certificate authority from the **CA Name** drop-down list, then click **Edit**.
3. Select the certificate policy you want to configure in the **Available Certificates** list.
4. Set the following options:
   - **Archive Keys** – select the appropriate SecureVault server.

     You can use different SecureVault servers for different certificate policies; for example, if you have three certificate policies, you could archive the encryption certificate in **Secure Vault**, the authentication certificate in **Secure Vault 2**, and leave the signing certificate unarchived.

     The **Certificate Authorities** workflow lists all ten possible instances of MyID SecureVault, whether or not you have configured them in the **External Systems** workflow; make sure you use the correct instance to match the instance you have already configured.

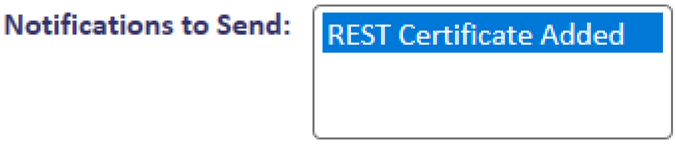
     Note: Support for multiple instances of MyID SecureVault is available with MyID CMS 12.16 or later. Earlier versions of MyID CMS supported only a single instance of SecureVault with a fixed name; however, updates are available for MyID CMS 12.13 and 12.14 to add support for multiple instances of MyID SecureVault.
   - **External Notification Data** – type the object ID of the certificate policy on the MyID CMS for Key Escrow system under which you want to import the certificate.

     You can find the object ID of the certificate policy in the database of the MyID CMS for Key Escrow system; see section *B.3.3*, *Setting up your certificate policy*.
   - **External System Notifications** – select the notifications you want to trigger when a certificate using this policy is added to the system. The box lists all of the external systems you have set up with a **Notification** type of **REST Certificate Added**. If you want to trigger the notification to multiple systems, hold CTRL or SHIFT and click to select multiple items in the list.

     Make sure the required notifications are highlighted in the list:

     To deselect a notification, hold CTRL and click.

     Note: For MyID 12.13 systems, this setting is named **Notifications to Send**, and contains only the **REST Certificate Added** notification. Make sure the **REST Certificate Added** notification is highlighted in the list:



Notifications to Send: REST Certificate Added

5. Click **Save**.

B.4.4 Restricting access to key recovery workflows

Because you want to carry out your key recovery operations only on the MyID CMS for Key Escrow system, you must restrict access to the key recovery workflows. You can do this using the **Edit Roles** workflow, or if you want to prevent the features from being available in the **Edit Roles** workflow at all, you can disable the operations in the MyID CMS database.

Intercede can provide you with a database script to disable your key recovery operations on your production MyID CMS system.

Appendix C Configuring MyID CMS for biometric storage

You can configure MyID CMS to store its biometric data in MyID SecureVault. Once you have configured MyID CMS to use MyID SecureVault to store its biometric data, whenever you enroll biometric data into MyID CMS, instead of storing it in the MyID CMS database, it sends it to MyID SecureVault, where it is encrypted and stored in the MyID SecureVault database. If you have configured MyID SecureVault to use an HSM, the HSM is used to encrypt the data. Whenever MyID CMS needs to access the data, it retrieves it from MyID SecureVault.

MyID CMS supports storing the following types of biometric data in MyID SecureVault:

- Fingerprints, including 10-slap biometrics.
- Facial biometrics.
- Iris biometrics.

You can also migrate your existing biometric data to MyID SecureVault using a command-line batch utility.

This chapter contains the following information:

- Prerequisites for using MyID CMS to store biometric data in MyID SecureVault.

  See section *C.1*, *Prerequisites for biometric storage*.
- Configuring MyID CMS to store biometric data in MyID SecureVault.

  See section *C.2*, *Configuring the options for biometric storage*.
- Migrating biometric data from MyID CMS to MyID SecureVault using a command-line utility.

  See section *C.3*, *Migrating biometric data to MyID SecureVault*.
- Details of reporting and auditing the biometric data stored in MyID SecureVault.

  See section *C.4*, *Reporting and auditing biometric usage*.

C.1 Prerequisites for biometric storage

To set up your environment to store biometric data from MyID CMS in MyID SecureVault, you require the following:

- An installation of MyID CMS PIV 12.17 or later.

  If you want to implement this system on a MyID CMS PIV 12.13 system, you must install the following update:

  - `UPDATE-12.13.0.2` – this update adds support for storing biometric data in MyID SecureVault as well as support for multiple instances of MyID SecureVault.

    Note: MyID CMS PIV 12.13 does not support the use of EFT files (which contain fingerprint biometrics) when integrating with MyID SecureVault. If your MyID CMS 12.13 system is configured to use EFT files, this causes errors. You must make sure the **Capture EFT Biometric Samples** configuration option (in the **Biometrics** section of the MyID CMS Settings) is set to `No`, if it exists – this option appears only on systems that have been updated with additional modules that provide EFT support.

  Note: MyID Enterprise does not support the capture or storage of biometric data. You must use a MyID PIV system.

- An installation of MyID SecureVault.

  Biometric storage is available in MyID SecureVault 3.0.0 and later.

  You must configure MyID CMS to integrate with MyID SecureVault, including setting up the external system with details of authentication. See section *A.2*, *Setting up the MyID SecureVault external system*.

- Optionally, an HSM.

  If you have an HSM configured for your MyID SecureVault system, it is used to encrypt the biometric data stored in the MyID SecureVault database.

Note: MyID CMS does not require any third-party biometric software to transfer biometric data between MyID CMS and MyID SecureVault.

C.2 Configuring the options for biometric storage

Important: Make sure you set up your connection to MyID SecureVault before configuring the options for biometric storage. If the connection to MyID SecureVault is not available when you enroll biometric data, this causes errors. See section *A.2*, *Setting up the MyID SecureVault external system* for details of setting up the connection.

To set up the storage of biometric data in MyID SecureVault, you must specify which instance of MyID SecureVault you want to use to store your biometric data. MyID CMS supports up to ten instances of MyID SecureVault, allowing you to use a separate instance of MyID SecureVault to store your biometric data than the one being used to store your keys, if required.

To configure MyID CMS to store biometric data in MyID SecureVault:

1. In MyID Desktop, from the **Configuration** category, select **Operation Settings**.

   From MyID CMS 12.16, use the **Configuration > Settings** feature in the MyID Operator Client instead.
2. In the **Biometrics** section, set the following:
   - **Biometric storage location** – set this to the location where you want to store your biometric data.

     Select one of the following:
     - **Internal** – store the biometric data in the MyID CMS database.

       This is the default option.
     - **SecureVault**, **SecureVault2** to **SecureVault10** – store the biometric data in the corresponding instance of MyID SecureVault as configured in the **External Systems** workflow; see section *A.2*, *Setting up the MyID SecureVault external system*.
3. Save your changes.

Important: Setting this option does not transfer any existing biometric data. Any attempts to retrieve existing data will fail. You must use the Biometric Migration Utility to transfer your existing biometric data from MyID CMS to MyID SecureVault; see section *C.3*, *Migrating biometric data to MyID SecureVault*.

Note, however, that once transferred to MyID SecureVault, you *cannot* migrate your biometric data back to MyID CMS or to another instance of MyID SecureVault, which means that changing the **Biometric storage location** option from a MyID SecureVault instance back to **Internal** or a different MyID SecureVault instance prevents you from retrieving your biometric data.

C.3 Migrating biometric data to MyID SecureVault

Once you have configured MyID CMS to store its biometric data in MyID SecureVault, the Biometric Migration Utility allows you to transfer your existing biometric data from MyID CMS to MyID SecureVault.

Important: You cannot transfer biometric data from MyID SecureVault back into MyID CMS, or to another instance of MyID SecureVault.

For assistance with biometric data migration, contact Intercede support quoting reference SUP-414.

C.4 Reporting and auditing biometric usage

The `Audits` table in the MyID SecureVault audit database contains auditing details of the biometric operations carried out in MyID SecureVault.

When MyID CMS stores biometric data in MyID SecureVault, it provides a unique identifier for the person (from the `ObjectID` field in the `UserAccounts` table, which is also visible in the URL bar of the browser when using View Person in the MyID Operator Client). This is recorded in the `ExternalLinkID` field of the `Audits` table.

See section *6.3*, *Auditing* for more information about auditing.

The SecureVault Biometric Usage report is also available in the MyID Operator Client, providing details of the number of records stored in MyID SecureVault; see section *A.6.2*, *SecureVault Biometric Usage report*.