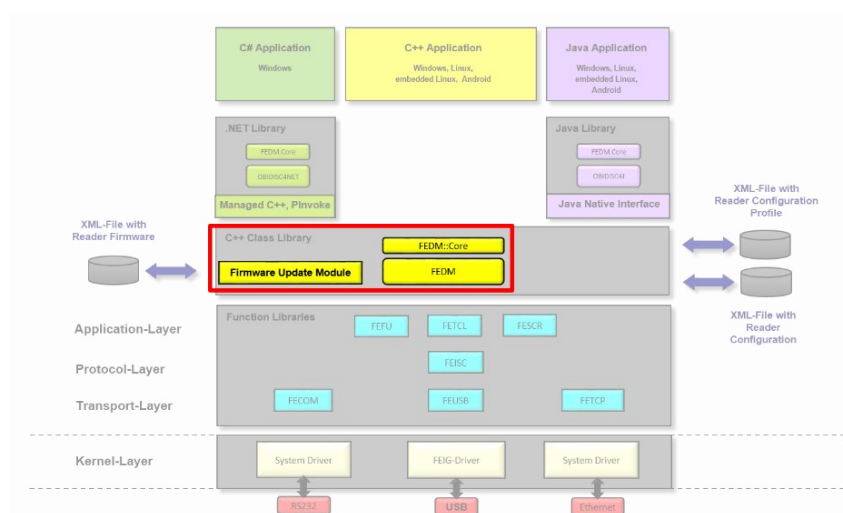


INSTALLATION MANUAL

ID ISC.SDK.xxx for C++



Operating System	Target		SDK Name	Notes
	32-Bit	64-Bit		
Windows Vista / 7 / 8 / 10	X	X	ID ISC.SDK.Win	
Linux	X	X	ID ISC.SDK.Linux	Other than Intel CPU on request
Raspberry Pi 3 and 4	X	-	ID ISC.SDK.Raspi	Raspbian OS (Stretch and Buster)
Android	X	-	ID ISC.SDK.Android	On request

Note

© Copyright 2017-2020 by FEIG ELECTRONIC GmbH
Lange Straße 4
D-35781 Weilburg
Germany
eMail: identification-support@feig.de

This manual supercedes all previous editions.
The information contained in this manual is subject to change without notice.

Copying of this document, and giving it to others and the use or communication of the contents thereof, is forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved in the event of the grant of a patent or the registration of a utility model or design.

The information contained in this manual has been gathered with all due care and to the best of our knowledge. FEIG ELECTRONIC GmbH assumes no liability for the accuracy and completeness of the data in this manual. In particular, FEIG ELECTRONIC GmbH cannot be held liable for consequential damages resulting from inaccurate or incomplete information. Since even with our best efforts this document may still contain mistakes, please contact us should you find any errors.

FEIG ELECTRONIC GmbH assumes no responsibility for the use of any information contained in this manual and makes no representation that they free of patent infringement. FEIG ELECTRONIC GmbH does not convey any license under its patent rights nor the rights of others.

The installation instructions given in this manual are based on advantageous boundary conditions. FEIG ELECTRONIC GmbH does not give any guarantee promise for perfect function of an Identification-system in cross surroundings.

Licensing agreement for use of the software

This is a contract between you and FEIG ELECTRONIC GmbH (hereinafter called "FEIG") concerning the use of the provided software **ID ISC.SDK.xxx for C++** (application programs, program libraries, source code examples and documents), hereafter called licensing material. By installing and using the software you agree to all terms and conditions of this agreement without exception and without limitation. If you are not or not completely in agreement with the terms and conditions, you may not install the licensing material or use it in any way. This licensing material remains the property of FEIG ELECTRONIC GmbH and is protected by international copyright.

§1 Subject and Scope of the Agreement

1. FEIG grants you the right to install the Software and to use it under the conditions specified below.
2. The licensed material is intended for use by an individual developer (single user license). You may install all the components of the licensed material on a hard-disk of a single computer which is intended for your use.
3. Installation and use may also include a network fileserver as long as the use is exclusive to the licensee. A separate license is required for any additional user.
4. You may make a backup copy of the licensed material.
5. FEIG grants you the right to use the documented C/C++ libraries for developing your own application program and to sell this C/C++ libraries only together with your application programs without payment of licensing fees so long as these application programs are used only to control or operate devices and/or systems which are developed and/or sold by FEIG.
6. FEIG grants you the right to use and modify the source code of the supplied program examples for developing your own application programs and to sell these application programs together with the C/C++ libraries without payment of licensing fees so long as these application programs are used only to control or operate devices and/or systems which are developed and/or sold by FEIG.
7. This license material can depend on third-party software. In case of the use of this third-party software the listed license agreements in chapter [Third-party Licensing agreements](#) have to be applied.

§2 Protection of the Licensed Material

1. The licensed material is the intellectual property of FEIG and its suppliers, and its structure, organization and code are the valuable trade secrets of FEIG and its suppliers. The licensed material is also protected by German copyright law, International Treaty provisions and the laws of the country in which it is used.
2. You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the executable application programs, program libraries and documents.
3. To the extent that FEIG has applied protections such as copyrights and other legal restrictions, you are required to retain these without modification and to incorporate them in unmodified form into all complete or partial copies which you produce.
4. The publication and transmission to third parties of licensed material prohibited as long as no explicit agreement to the contrary has been established between you and FEIG. This provision does not apply to such application programs as have been created and sold under §1 Par. 5 and Par. 6 of this Agreement.

§3 Warranty and Limitation of Liability

1. You agree with FEIG that it is not possible to develop electronic data processing programs such that they are without defect for all application conditions. FEIG calls explicit attention to the fact that the installation of a new program may affect already existing software, including software which does not run simultaneous with the new software. In no event

will FEIG be liable to you for any consequential, incidental or special damages, including any lost profits or lost savings. If you want to be sure that no already installed program will be affected, you may not install the licensed material.

2. FEIG calls explicit attention to the fact that installing the licensed material may result in irreversible settings and adjustments to devices which may in turn destroy or otherwise make them unusable. FEIG assumes no liability for such actions whether knowingly or unknowingly.

3. No Warranty. The Software is being delivered to you AS IS and FEIG makes no warranty as to its use or performance. FEIG makes no warranties, express or implied, as to noninfringement of third party rights, merchantability or fitness for any particular purpose.

4. FEIG call explicit attention the licensed material is not designed with components and testing for a level of reliability suitable for use in or in connection with surgical implants or as critical components in any life support systems whose failure to perform can reasonably be expected to cause significant injury to a human.

To avoid damage, injury, or death, the user or application designer must take reasonably prudent steps to protect against system failures.

§4 Other

1. The parties agree that this constitutes the sole and entire agreement of the parties as to the matter set forth herein and supersedes any previous agreements, understandings, and arrangements between the parties relating hereto. Any modifications or additions must be made in written form.

2. If any provision in this Licensing Agreement should be held illegal or unenforceable by a court having jurisdiction, such provision shall be modified to the extent necessary to render it enforceable without losing its intent, or severed from this Licensing Agreement if no such modification is possible, and other provisions of this Licensing Agreement shall remain in full force and effect.

3. This License Agreement shall be construed, interpreted, and governed by the laws of the Federal Republic of Germany and the venue of any legal action against FEIG shall be Weilburg.

Please refer any questions concerning these Agreements to:

FEIG ELECTRONIC GmbH

Lange Straße 4

35781 Weilburg

Tel.: 06471 / 31090

Fax: 06471 / 310999

E-Mail: identification-support@feig.de

Internet: <http://www.feig.de>

Third-party Licensing agreements

Licensing agreement of openSSL organization

The following license issues are to be applied in the case that encrypted data transmission is used.

LICENSE ISSUES
=====

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

=====
Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. 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.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED 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 OpenSSL PROJECT OR ITS 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.
=====

This product includes cryptographic software written by Eric Young (eyay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eyay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eyay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used.

This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

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

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

2. 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.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

"This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"

The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:

"This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``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 AUTHOR 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.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Contents

Licensing agreement for use of the software.....	3
Third-party Licensing agreements	5
Licensing agreement of openssl organization	5
Contents	7
1. Revision history	9
2. Introduction	10
3. Installation	11
3.1. 32- and 64-Bit Windows Vista / 7 / 8 / 10	11
3.1.1. Lib and DLL files.....	12
3.2. 32- and 64-Bit Linux and Raspberry Pi	13
3.2.1. Binary files	14
3.3. Dependencies	14
4. Integration into application projects	15
4.1. Supported Development Tools.....	15
4.2. Incorporating into Visual Studio using the Qt-Integration	15
4.3. Incorporating into Qt-Creator	16
4.4. Incorporating into Visual Studio manually	16
5. Installation on the target computer	20
5.1. 32-Bit Libraries on a 32- and 64-Bit Windows	20
5.2. 64-Bit Libraries on a 64-Bit Windows.....	21
5.3. 32- and 64-Bit Linux	22

Remarks Concerning the Documentation

FEIG ELECTRONIC GmbH does not duplicate information about FEIGr readers in different manuals or include cross-references to certain page numbers of another document. This is because the manuals are constantly updated, and helps to eliminate mistakes resulting from information obtained from out-of-date documents. We therefore encourage the user of this library to always verify that he is using the current manuals. The newest versions can always be obtained from FEIG ELECTRONIC GmbH.

Important notes:

You are only permitted to use this library if you have first agreed to the license conditions on the back of this page.

Anyone is free to modify source code. Therefore you should work only with libraries you have received directly from FEIG ELECTRONIC GmbH. In any case further transmission of the source code is prohibited.

1. Revision history

Revision	Date	Changes
3	09.07.2020	<ul style="list-style-type: none">• Support for Raspberry Pi 3 Stretch and Buster added• Support for Raspberry Pi 4 Buster added• Support for Raspberry Pi 2 removed• Support for Raspberry Jessie operating system removed
2	27.07.2018	<ul style="list-style-type: none">• Support for Raspberry Pi added• Support for Visual Studio 2017
1	08.03.2018	<ul style="list-style-type: none">• Remove of fedm-myaxxess library• For Windows: add of information regarding the libraries with C API• For Windows: remove of MFC dependency for C++ libraries built with Visual Studio 2013• For Linux: add of support for LogManager
0	25.09.2017	<ul style="list-style-type: none">• Initial version

2. Introduction

FEIG ELECTRONIC GmbH has developed different, hierarchical structured software libraries to simplify the integration of FEIG RFID readers into customers applications.

A common attribute of all components is the support of all FEIG reader families with a uniform Application Programming Interface (API).

The class library for C++ represents the highest level in the software stack and supports all FEIG reader families.

This class library is split into an implementation layer (**FedmCore**) and an API layer (**FedmCoreApi**). The first one is established since the year 2000 and can also be used as an API layer, but since 2017 it is wrapped by a new and modern API layer above the implementation layer and it is recommended to use this API for new projects.

After installation, you should start with the Tutorial (H60810-0e-ID-B) and with the C++ samples of the SDK.

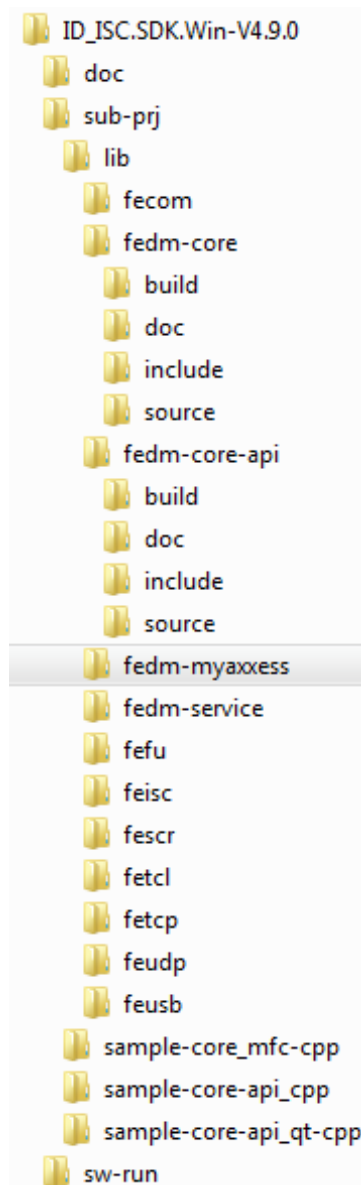
Further information can also be found in the System Manual of your preferred FEIG RFID reader.

3. Installation

Normally, this package is shipped together with other libraries in a Software Development Kit (SDK). Copy the SDK into a directory of your choice.

The files of this library package can be found in the sub-directories fedm-core, fedm-core-api and fedm-service.

3.1. 32- and 64-Bit Windows Vista / 7 / 8 / 10



LIB files are located under

"sub-prj\lib\fe<library>\build\windows\..."

DLL files are located under "sw-run\..."

Include files are located under

"sub-prj\lib\fe<library>\include"

The fedm-libraries (core, core-api, service) have additional include files under

"sub-prj\lib\fedm<library>\source\..."

The relative directory structure of all fedm-includes must be maintained!

If you won't add your projects to the SDK path, we recommend the following steps:

- Copy all required DLL files from the SDK directories into the directory of the application program.
- Use all required LIB files from the SDK directories (recommended) or copy all required LIB files in the project or LIB directory.
- Use all required Include Directories from the SDK directories (recommended) or copy all required Header files into the project or INCLUDE path.

3.1.1. Lib and Dll files

We provide all libraries as 32 and 64 bit versions.

The low level libraries (fecom, feusb, fetcp, feudp, feisc, fefu, fescr, fetcl) have a standard C API.

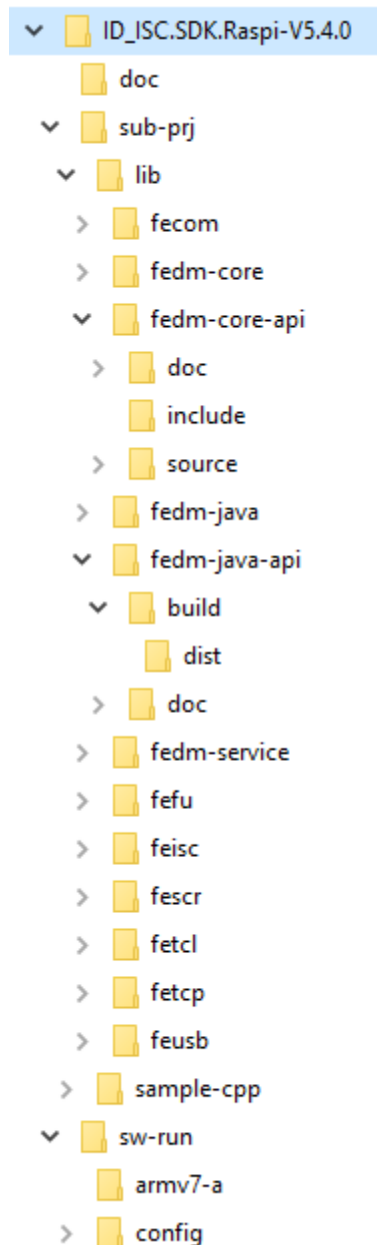
These libraries contain all functions you need to develop any RFID application.

Additionally we provide the high level libraries (fedm-core, fedm-service and the new fedm-core-api) with a C++ API.

The SDK contains versions build with Microsoft Visual Studio 2012 (vc110) and Microsoft Visual Studio 2013 (vc120).

Build with IDE	Library files located in	Description
Visual Studio 2012 (vc110)	sw-run\x86\vc110	C Libraries FECOM, FEUSB, FETCP, FEISC and FETCL depend NOT on Windows.h
	sw-run\x64\vc110	C++ Libraries fedm-core, fedm-core-api and fedm-service depend NOT on MFC.
Visual Studio 2013 (vc120)	sw-run\x86\vc120	C Libraries FECOM, FEUSB, FETCP, FEISC and FETCL depend NOT on Windows.h
	sw-run\x64\vc120	C++ Libraries fedm-core, fedm-core-api and fedm-service depend NOT on MFC.
Visual Studio 2017 (vc141)	sw-run\x86\vc141	C Libraries FECOM, FEUSB, FETCP, FEISC and FETCL depend NOT on Windows.h
	sw-run\x64\vc141	C++ Libraries fedm-core, fedm-core-api and fedm-service depend NOT on MFC.

3.2. 32- and 64-Bit Linux and Raspberry Pi



The binaries are under "sw-run/x86" and "sw-run/x64".
Choose one option for installation:

Option 1: Run the install script:

```
"install.sh <destdir>"
```

It copies all library files into the destination directory and creates symbolic links for each library file and finally calls ldconfig.

Option 2: Copy all files under "sw-run" into a directory of your choice and create symbolic links for all lib...so.x.y.z in the destination with the following calls:

```
cd <destdir>
```

```
ln -s /< destdir >/lib...so.x.y.z lib...so.x
```

```
ln -s /< destdir >/lib...so.x lib...so
```

```
ldconfig
```

x.y.z represents the version number of the library

Include files are located under

"sub-prj/lib/fe<library>/include"

The fedm-libraries (core, core-api, service) have additional include files under

"sub-prj/lib/fedm<library>/source/..."

The relative directory structure of all fedm-includes must be maintained!

3.2.1. Binary files

We provide all libraries as 32 and 64 bit versions.

The low level libraries (fecom, feusb, fetcp, feudp, feisc, fefu, fescr, fetcl) have a standard C API and contain all functions you need to develop any RFID application.

Additionally we provide the high level libraries (fedm-core, fedm-service and the new fedm-core-api) with a C++ API.

The compiled libraries are linked against LibC V6 und LibStdC++ V6.

3.3. Dependencies

The FedmlscCoreApi class library depends on the implementation layer FedmlscCore, the service library FedmlscService with support for Firmware Update, the function libraries for the communications interfaces (FECOM, FEUSB, FETCP) and the reader family (FEISC, FEFU, FETCL). These libraries are part of the respective SDK and must be installed on the target system.

The class Utility::LibInfo contains a method *EvalLibDependencies* for the verification of the version numbers of the dependent function libraries. It is recommended to invoke this method once in the application after the program started.

4.Integration into application projects

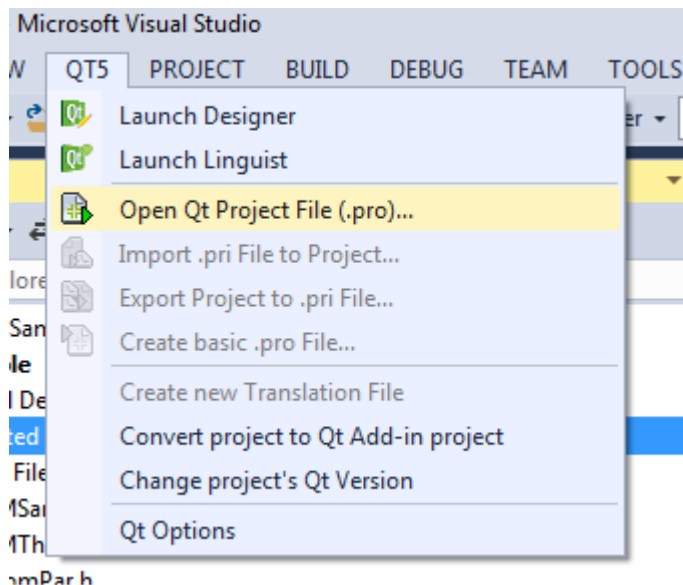
4.1.Supported Development Tools

Operating System	IDE	Support
Windows / Vista / 7 / 8 / 10	Visual Studio 2012 / 2013	VS2012 and VS2013 with Professional Version
Linux	GCC up from v5	yes
RaspBerry Pi	GCC up from v5	yes
Android	On request	On request

4.2. Incorporating into Visual Studio using the Qt-Integration

Modern project management is controlled by profiles. Profiles describe projects in a platform and toolchain independent way.

When developing an application with Qt, one of the first steps is to create a profile. The SDK contains 3 Qt-samples (path "sub-prj\sample-core-api_qt-cpp"). With the Qt-Integration for Visual Studio you load the profile:



For the BRMSample open

```
"sub-prj\sample-core-api_qt-cpp\BRMSample\build\windows\x86\vc120\
BRMSample.pro"
```

This will generate the entire Visual Studio project.

Under the same directory we also provide the complete solution for that project.

4.3. Incorporating into Qt-Creator

Under Windows and Linux you can directly open the profile with Qt-Creator. Yust open

```
"sub-prj\sample-core-api_qt-cpp\BRMSample\build\BRMSample.pro"
```

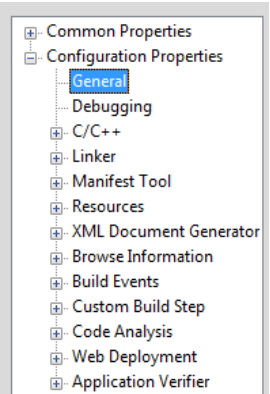
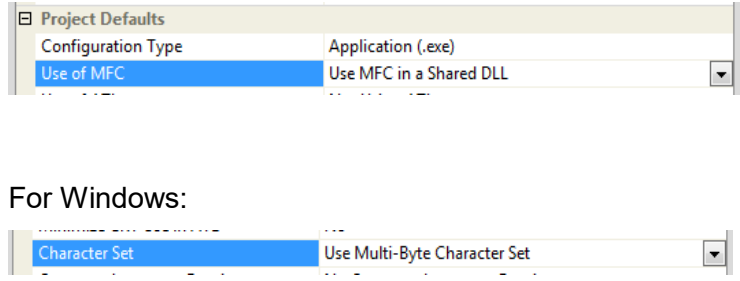
(Notice: this is another path)

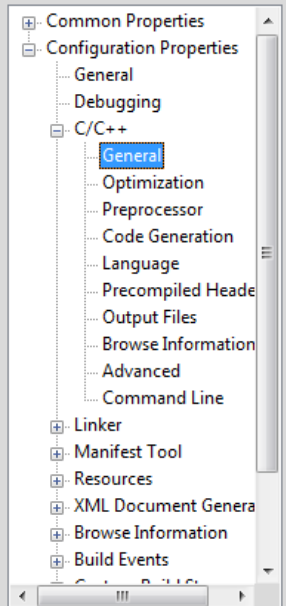
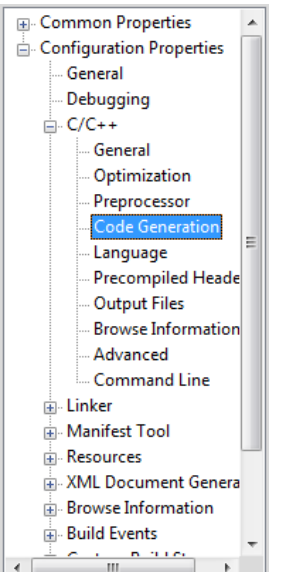
4.4. Incorporating into Visual Studio manually

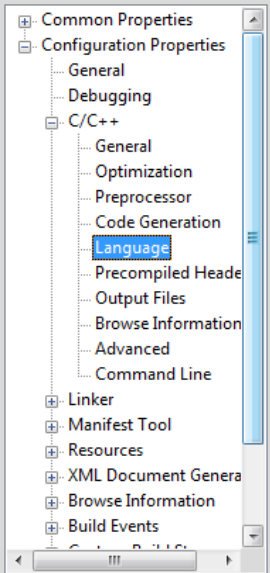

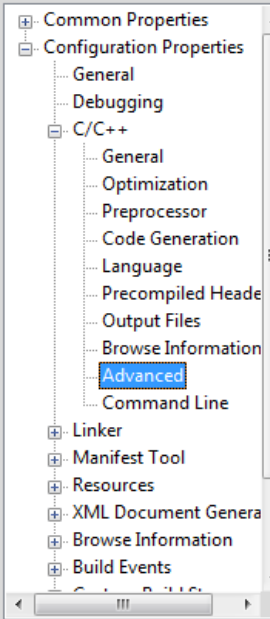
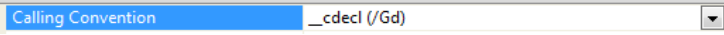
To build a Visual Studio project manually, consider the MFC-Samples under

```
"sub-prj\sample-core_mfc-cpp"
```

Important settings for Visual Studio projects:

Project Group	Property Setting
	

Project Group	Property Setting																												
	<p>Add all required directories for Header files.</p> <table border="1"> <tr> <td>Additional Include Directories</td><td><code>\$(SolutionDir)..\..\source;\$(SolutionDir)..\vc\$(PlatformT</code></td></tr> <tr> <td>Additional #using Directories</td><td></td></tr> <tr> <td>Debug Information Format</td><td>Additional Include Directories</td></tr> <tr> <td>Common Language RunTime Support</td><td></td></tr> <tr> <td>Consume Windows Runtime Extension</td><td></td></tr> <tr> <td>Suppress Startup Banner</td><td><code>\$(SolutionDir)..\..\source</code></td></tr> <tr> <td>Warning Level</td><td><code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)</code></td></tr> <tr> <td>Treat Warnings As Errors</td><td><code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)\resource\res</code></td></tr> <tr> <td>SDL checks</td><td><code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)\resource</code></td></tr> <tr> <td>Multi-processor Compilation</td><td><code>\$(SolutionDir)..\..\lib\fedm-core\include</code></td></tr> <tr> <td></td><td><code>\$(SolutionDir)..\..\lib\fecom\include</code></td></tr> <tr> <td></td><td><code>\$(SolutionDir)..\..\lib\feisc\include</code></td></tr> <tr> <td></td><td><code>\$(SolutionDir)..\..\lib\fetcp\include</code></td></tr> <tr> <td></td><td><code>\$(SolutionDir)..\..\lib\feusb\include</code></td></tr> </table>	Additional Include Directories	<code>\$(SolutionDir)..\..\source;\$(SolutionDir)..\vc\$(PlatformT</code>	Additional #using Directories		Debug Information Format	Additional Include Directories	Common Language RunTime Support		Consume Windows Runtime Extension		Suppress Startup Banner	<code>\$(SolutionDir)..\..\source</code>	Warning Level	<code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)</code>	Treat Warnings As Errors	<code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)\resource\res</code>	SDL checks	<code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)\resource</code>	Multi-processor Compilation	<code>\$(SolutionDir)..\..\lib\fedm-core\include</code>		<code>\$(SolutionDir)..\..\lib\fecom\include</code>		<code>\$(SolutionDir)..\..\lib\feisc\include</code>		<code>\$(SolutionDir)..\..\lib\fetcp\include</code>		<code>\$(SolutionDir)..\..\lib\feusb\include</code>
Additional Include Directories	<code>\$(SolutionDir)..\..\source;\$(SolutionDir)..\vc\$(PlatformT</code>																												
Additional #using Directories																													
Debug Information Format	Additional Include Directories																												
Common Language RunTime Support																													
Consume Windows Runtime Extension																													
Suppress Startup Banner	<code>\$(SolutionDir)..\..\source</code>																												
Warning Level	<code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)</code>																												
Treat Warnings As Errors	<code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)\resource\res</code>																												
SDL checks	<code>\$(SolutionDir)..\vc\$(PlatformToolsetVersion)\resource</code>																												
Multi-processor Compilation	<code>\$(SolutionDir)..\..\lib\fedm-core\include</code>																												
	<code>\$(SolutionDir)..\..\lib\fecom\include</code>																												
	<code>\$(SolutionDir)..\..\lib\feisc\include</code>																												
	<code>\$(SolutionDir)..\..\lib\fetcp\include</code>																												
	<code>\$(SolutionDir)..\..\lib\feusb\include</code>																												
	<p>for Release Version:</p> <table border="1"> <tr> <td>Runtime Library</td><td>Multi-threaded DLL (/MD)</td></tr> </table> <p>for Debug Version:</p> <table border="1"> <tr> <td>Runtime Library</td><td>Multi-threaded Debug DLL (/MDd)</td></tr> </table>	Runtime Library	Multi-threaded DLL (/MD)	Runtime Library	Multi-threaded Debug DLL (/MDd)																								
Runtime Library	Multi-threaded DLL (/MD)																												
Runtime Library	Multi-threaded Debug DLL (/MDd)																												

Project Group	Property Setting
	
	

The image shows the Visual Studio interface with two tabs: 'Project Group' and 'Property Setting'. In the 'Project Group' tab, the 'Linker' > 'General' section is expanded, and the 'Input' property is selected. In the 'Property Setting' tab, the 'Additional Dependencies' list is populated with the following paths:

- \$(SolutionDir)..\..\..\lib\fecom\build\windows\\$(PlatformShortName)\vc\$(PlatformToolsetVersion)\\$(Configuration)\fecom.lib
- \$(SolutionDir)..\..\..\lib\feisc\build\windows\\$(PlatformShortName)\vc\$(PlatformToolsetVersion)\\$(Configuration)\feisc.lib
- \$(SolutionDir)..\..\..\lib\fetcp\build\windows\\$(PlatformShortName)\vc\$(PlatformToolsetVersion)\\$(Configuration)\fetcp.lib
- \$(SolutionDir)..\..\..\lib\feusb\build\windows\\$(PlatformShortName)\vc\$(PlatformToolsetVersion)\\$(Configuration)\feusb.lib
- \$(SolutionDir)..\..\..\lib\fedm-core\build\windows\\$(PlatformShortName)\vc\$(PlatformToolsetVersion)\\$(Configuration)\FedmlscCoreVC\$(PlatformToolsetVersion).dll

5. Installation on the target computer

Together with the application files, the runtime file of the library FedmlscCore, FedmlscCoreApi, FedmlscService and the runtime files of the function libraries FECOM, FEUSB, FETCP, FEISC, FETCL and FEFU must be installed on the target computer.

5.1. 32-Bit Libraries on a 32- and 64-Bit Windows

It is recommended to keep the library files in the directory of the application. This avoids version conflicts with later installations which also install these library files, but possibly different versions.

The library files FedmlscCoreVC1xx.dll, FedmlscCoreApiVC1xx.dll and FedmlscServiceVC1xx.dll depend on newer C/C++ runtime libraries which are usually not present on the target computer. Therefore, they must be installed. So-called Merge Modules are provided with Visual Studio which can be incorporated in a Setup project and which install the MFC libraries. The following table lists for each FedmlscCore library file the Merge Module to be installed to.

Library File	MFC Version	Merge Modules
FedmlscCoreVC110.dll	Version 11.0	Microsoft_VC110_MFC_x86.msm
FedmlscCoreApiVC110.dll	(11.0.51106.1)	Microsoft_VC110_CRT_x86.msm
FedmlscServiceVC110.dll		
FedmlscCoreVC120.dll	No dependency	Microsoft_VC120_CRT_x86.msm
FedmlscCoreApiVC120.dll		
FedmlscServiceVC120.dll		

Alternatively, the installation of the Visual C++ Runtime Libraries can be realized with the download site of Microsoft. For each version of MFC you can find a file called vcredist_x86.exe for download.

1st Note: The file vcredist_x86.exe must be of version of at least the specified number above.

2nd Note: Merge Modules can only be updated with Windows Update.

3rd Note: Debug versions of the C++ libraries, marked with a **d** at the end of the file name (e.g. FedmlscCoreApiVC110**d**.dll) cannot be installed on target computers. The reason is that every merge module does install only the release version of a MFC library.

5.2. 64-Bit Libraries on a 64-Bit Windows

It is recommended to keep the library files in the directory of the application. This avoids version conflicts with later installations which also install these library files, but possibly different versions.

The library files FedmlscCoreVC1xx.dll, FedmlscCoreApiVC1xx.dll and FedmlscServiceVC1xx.dll depend on newer C/C++ runtime libraries which are usually not present on the target computer. Therefore, they must be installed. So-called Merge Modules are provided with Visual Studio which can be incorporated in a Setup project and which install the MFC libraries. The following table lists for each FedmlscCore library file the Merge Module to be installed to.

Library File	MFC Version	Merge Modules
FedmlscCoreVC110.dll	Version 11.0	Microsoft_VC110_MFC_x64.msm
FedmlscCoreApiVC110.dll	(11.0.51106.1)	Microsoft_VC110_CRT_x64.msm
FedmlscServiceVC110.dll		
FedmlscCoreVC120.dll	No dependency	Microsoft_VC120_CRT_x64.msm
FedmlscCoreApiVC120.dll		
FedmlscServiceVC120.dll		

Alternatively, the installation of the Visual C++ Runtime Libraries can be realized with the download site of Microsoft. For each version of MFC you can find a file called vcredist_x64.exe for download.

1st Note: The file vcredist_x64.exe must be of version of at least the specified number above.

2nd Note: Merge Modules can only be updated with Windows Update.

3rd Note: Debug versions of C++ libraries, marked with a **d** at the end of the file name (e.g. FedmlscCoreApiVC110**d**.dll) cannot be installed on target computers. The reason is that every merge module does install only the release version of a MFC library.

5.3. 32- and 64-Bit Linux

The installation on the target is equivalent to [3.2. 32- and 64-Bit Linux and Raspberry Pi](#). But only the runtime files *.so may be installed on the target computer.

The installation instructions for the dependent function libraries can also be found in the respective manuals.