FabImage Library 5.3

Getting Started

Created: 7/20/2023

Product version: 5.3.4.94467

Table of content:

This is just a placeholder to silence warnings about broken link.

SDK Installation

Requirements

FabImage Library is designed to be a part of applications working under control of the Microsoft Windows operating system. Supported versions are: 7, 8 and 10, as well as the corresponding embedded editions.

To build an application using FabImage Library, Microsoft Visual Studio environment is required. Supported versions are: 2015, 2017 and 2019.

Running the Installer

The installation process is required to copy the files to the proper folders and to set the environment variables used for building applications using FabImage Library.

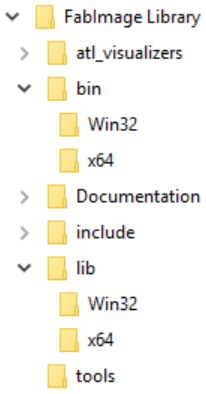
After the installation, a license for FabImage Library product has to be loaded. It can be done with the *License Manager* tool available in the *Start Menu*.

To verify that the installation has been successful and the license works correctly, one can try to load, build and run example programs, which are available from the *Start Menu*.

SDK Directories

FabImage Library is distributed as a set of header files (.h), dynamic (.dll) and static (.lib) libraries. The libraries (static and dynamic) are provided in versions for 32-bit and 64-bit system. The header files are common for both versions.

The picture below shows the structure of the directories containing headers and libraries included in FabImage Library.



The directories (installed in the *Program Files* system folder) being a part of FabImage Library are shortly described below.

- **ftl_visualizers** – a directory containing the visualizers for Microsoft Visual Studio Debugger of FabImage Library data types.
- **bin** – a directory containing dynamic linked library files (FIL.dll) for 32|64-bit applications. The libraries are common for all supported versions of Microsoft Visual Studio and for Debug|Release configurations. All the functions of FabImage Library are included in the FIL.dll file.
- **Documentation** – a directory containing the documentation of FabImage Library, including this document.
- **include** – a directory containing all header (.h) files for FabImage Library. Every source code file that uses FabImage Library needs the FIL.h header file (the main header file) to be included.
- **lib** – a directory containing static (.lib) libraries (FIL.lib) for 32|64-bit applications. The FIL.lib file has to be statically-linked into the program that uses FabImage Library. It acts as an intermediary between the usage of FabImage Library functions and the FIL.dll file. The programmer creating an application does not need to bother about DLL entry points and functions exported from the FIL.dll file. FabImage Library is designed to be easy to use, so one only needs to link the FIL.lib file and can use all the functions from the FIL.dll just as easy as local functions.
- **tools** – a directory containing the *License Manager* tool helping the user to load the license for FabImage Library to the developer's computer.
- **Examples** – a directory located in the *Public Documents* system folder (e.g. `C:\Users\Public\Documents\FabImage Library 5.3\Examples` on Windows Vista/7) containing simple example solutions using FabImage Library. The examples are a good way of learning, how to use FabImage Library. They can be used as a base for more complicated programs as well. The shortcut to the Examples directory can be found in the *Start Menu* after the installation of FabImage Library.

Library Architecture

FabImage Library is split into four parts:

1. FabImage Library - contains all functions for working with images.
2. Standard Library - contains all auxiliary functions like: file operations, XML editing or mathematical operations.
3. GenICam Library - contains all GenICam and GigEVision functions.
4. Third Party Library - contains functions of third-party hardware producers.

The usage of the library is possible only when including one of the following header files:

- FIL.h
- STD.h
- Genicam.h

- ThirdPartySdk.h

Environment and Paths

FabImage Library uses the environment variable named `FIL_PATH5_3` (5_3 stands for the 5.3 version) in the building process. The variable points the directory with the headers and libraries needed in the compile time (.h files and FIL.lib) and in the run time (FIL.dll). Its value is typically set to `C:\Program Files (x86)\FabImage\FabImage Library 5.3`, but it can differ in other systems.

The projects using FabImage Library should use the value of `FIL_PATH5_3` to resolve the locations of the header files and statically-linked `FIL.lib` file. Using an environment variable containing path makes the application source code more portable between computers. The `FIL_PATH5_3` path is typically used in the project settings of the compiler (Configuration Properties | C/C++ | General | Additional Include Directories) to find the header files, settings of the linker (Configuration Properties | Linker | General | Additional Library Directories) to find the proper version of the FIL.lib and in the configuration of Post-Build Event (Configuration Properties | Build Events | Post-Build Event | Command Line) to copy the proper version of the FIL.dll file to the output directory of the project. All the settings can be viewed in the simple example applications distributed with FabImage Library.

Project Configuration

General Information

FabImage Library is designed to be used as a part of C++ projects developed with Microsoft Visual Studio in versions 2015-2019.

Creating a New Project

Microsoft Visual Studio 2015, 2017 and 2019

FabImage Library is provided with a project template. To create a new project using FabImage Library, start Microsoft Visual Studio and choose the *File* | *New* | *Project...* command. The template called *FIL 5.3 Project* is available in the tab *Installed* | *Templates* | *Other Languages* | *Visual C++*.

Required Project Settings

All projects that use FabImage Library need some specific values of the compiler and linker settings. If you want to use the Library in your existing project or you are manually configuring a new project, please apply the settings listed below:

- **Configuration Properties | General**
  - **Character Set** should be set to `Use Unicode Character Set`.
- **Configuration Properties | C/C++**
  - **General**
    - **Additional Include Directories** should contain the `$(FIL_PATH5_3)\include\` path.
- **Configuration Properties | Linker**
  - **General**
    - **Additional Library Directories** should contain the proper path to directory containing the FIL.lib file. The proper path is `$(FIL_PATH5_3)\lib\$(PlatformName)\`.
  - **Input**
    - **Additional Dependencies** should contain `FIL.lib` file.
- **Configuration Properties | Build Events**
  - **Post-Build Event**
    - **Command Line** should contain `copy "$(FIL_PATH5_3)\bin\$(PlatformName)\FIL.dll" "$(OutDir)"` call. This setting is not mandatory, but the application using FabImage Library requires an access to the FIL.dll file and this is the easiest way to fulfill this requirement.

Including Headers

Every source code file that uses FabImage Library needs the `#include <FIL.h>` directive. A proper path to the FIL.h file is set in the settings of the compiler (described above), so there is no need to use the full path in the directive.

Distributing FabImage Library with Your Application

Once the application is ready, it is time for preparing a distribution package or an installer. There are two requirements that needs to be fulfilled:

- The final executable file of the application needs to have access to the proper version (used by *Win32* or *x64* configuration) of the FIL.dll file. Typically, the FIL.dll file should be placed in the same directory as the executable.
- The computer that the application will run on needs a valid license for the use of FabImage Library product. Licenses can be managed with the License Manager application, that is installed with FabImage Library Runtime package.
- A license file (*.fikey) can be also manually copied to the end user's machine without installing FabImage Library Runtime. It must be placed in a subdirectory of the *AppData* system folder. The typical location for the license file is `C:\Users\%USERNAME%\AppData\Local\FabImage\Licenses`. Remember that the license is valid per machine, so every computer that runs the application needs a separate license file.
- Alternatively to the (*.fikey) files we support USB Dongle licenses.

Using Library with CMake

Library ships with CMake configuration modules. It makes the project portable, and easy to compile for Windows, linux or Android. The minimum CMake version supported is 3.10 (for example shipped with Ubuntu bionic/18.04)

Quick Start

A simple template for `CMakeLists.txt` is presented below:

```
cmake_minimum_required(VERSION 3.10)
project(example)

find_package(
    FIL
    # for a specific version, uncomment the line below
    #5.3
    CONFIG
    REQUIRED
)

# copy binaries to build directory
copy_fil()

add_executable(
    # executable name
    example_exec
    # source files
    main.cpp
)

target_link_libraries(
    example_exec
    PUBLIC
    FIL
)

# install user executable
install(TARGETS example_exec)
# install ALL FIL libraries
install_fil()
```

One can also copy one of the CMake examples, and modify to your needs. For further cmake use refer to online documentation. Be aware that ubuntu 18.04 is the baseline distribution, so minimal CMake version is 3.10

Reference

package

CMake package is provided for windows installer and linux archive. Both should be usable after installation. Linux additionally ships with Android libraries. The library is only discoverable using `CONFIG` mode, so it's sensible to restrict `find_package` to that mode.

```
find_package(
    FIL
    # for a specific version, uncomment the line below
    #5.3
    CONFIG
    REQUIRED
)
```

On Android to use system installed FIL it is necessary to add `CMAKE_FIND_ROOT_PATH_BOTH` argument:

```
find_package(FIL CONFIG REQUIRED CMAKE_FIND_ROOT_PATH_BOTH)
```

Possible packages:

- FIL - full library
- FIL_Lite - lite library
- Weaver - deep learning inference library

install_fil

Install all FIL libraries when executing `make install` or `ninja install` or building `INSTALL` project in Visual Studio. It accepts a `LIB` argument to override default installation directory. It requires `find_package(FIL...)` call first.

```
find_package(FIL CONFIG REQUIRED)

install_fil()
```

By default it installs to `${CMAKE_INSTALL_PREFIX}/bin` on Windows and `${CMAKE_INSTALL_PREFIX}/lib` on Linux. When provided the LIB argument it installs to `${CMAKE_INSTALL_PREFIX}/${LIB_ARGUMENT}`

```
install_fil(LIB "fil_directory")
```

Possible variants:

- `install_fil()`
- `install_fil_lite()`
- `install_weaver()`

copy_fil

Copy all FIL libraries when compiling targets that depend on FIL to binary directory. By default it's `${CMAKE_BINARY_DIR}` or `${CMAKE_BINARY_DIR}/$<CONFIG>` on Windows. It requires `find_package(FIL...)` call first.

```
find_package(FIL CONFIG REQUIRED)

copy_fil()
```

Possible variants:

- `copy_fil()`
- `copy_fil_lite()`
- `copy_weaver()`

This article is valid for version 5.3.4