

Install FreeFrameExtended / MultiIDE

Description

- this is freeFrameExtended Plugin for VVVV
- it use cmake to build your projectFiles
- so your projectFiles and your SourceCode is clearly seperated
- you can build your projectFiles for different IDEs (ie. VisualStudio/CodeBlocks ...)

Requirements:

CMAKE 2.6 or grater:

- <http://www.cmake.org/cmake/resources/software.html>
- download and install

for OpenCV use

- download and install
- <http://sourceforge.net/projects/opencvlibrary/files/opencv-win/1.0/>

Configure CMake

- open [freeFrameExtended\source\CMakeLists.txt](#)

```
PROJECT(FreeFrameExt CXX C)

MESSAGE(STATUS "-----")
MESSAGE(STATUS "Configuring: ${PROJECT_NAME}")
MESSAGE(STATUS "-----")

CMAKE_MINIMUM_REQUIRED(VERSION 2.6 FATAL_ERROR)

# PATH TO OPENCV 1.0
SET(OPENCV1.0_PATH "C:/Program Files (x86)/OpenCV")

# PATH TO VVVV
SET(VVVV_PATH "E:/Vvvv-45beta25.1/")

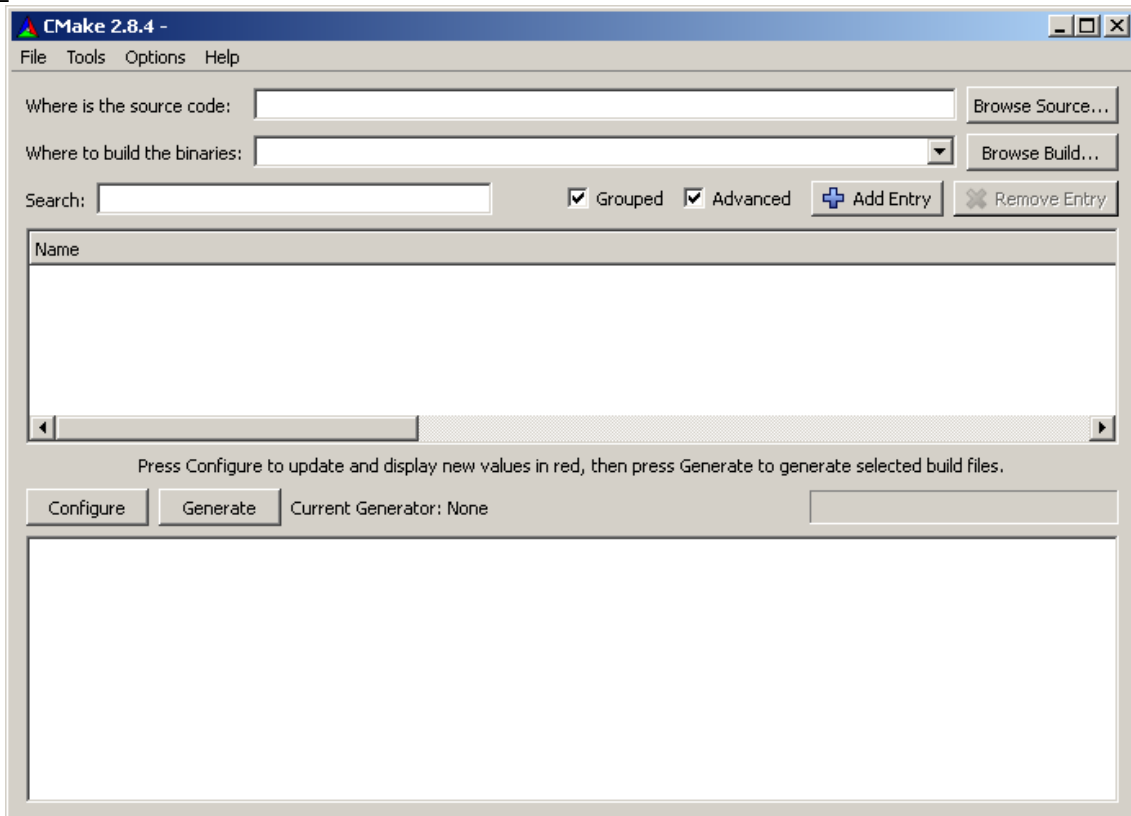
# IF TRUE/ COPY PLUGIN TO VVVV\freeframe
SET(COPY_TO_VVVV TRUE)

# DONT CAHNGE IS PATH TO REQUIRED CMAKE MODULES
SET(CMAKE_MODULE_PATH "${CMAKE_CURRENT_SOURCE_DIR}/_cmake")
INCLUDE(CommonConfig)
INCLUDE(MacroOptionalAddSubdirectory)

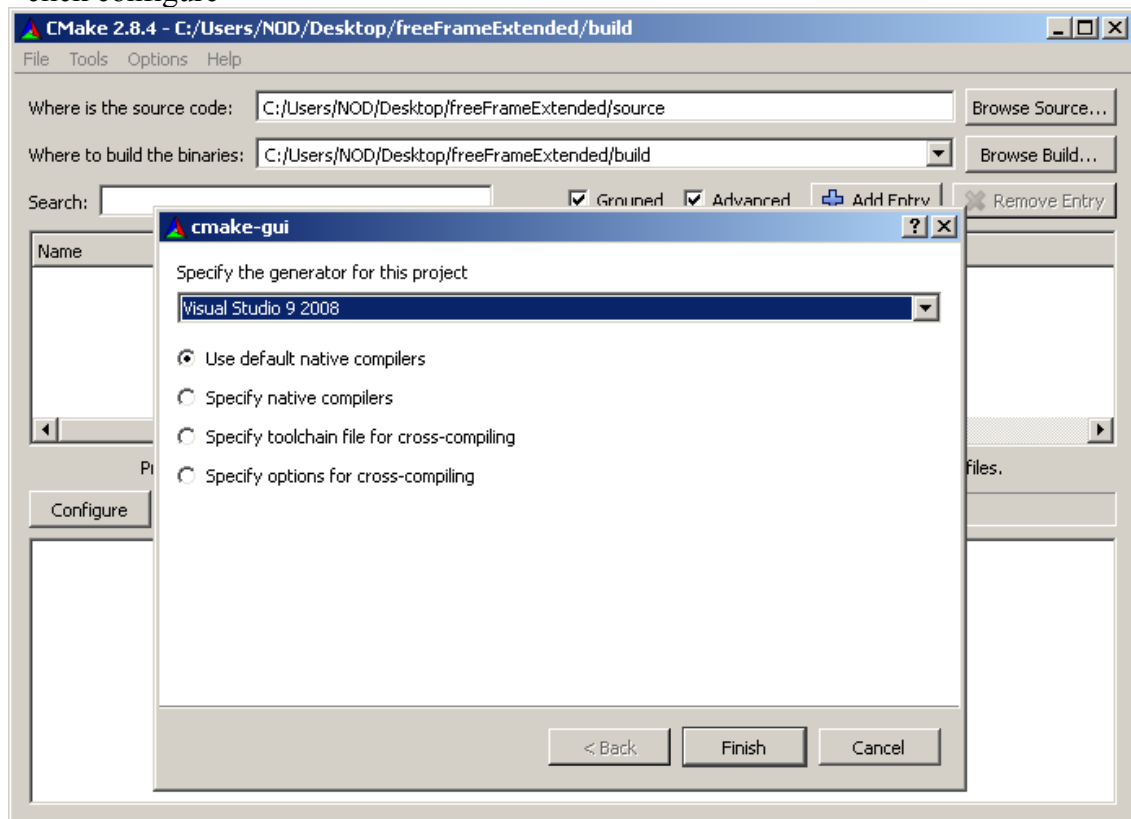
# ADD YOUR PROJECTS/SUBFOLDERS HERE
MACRO_OPTIONAL_ADD_SUBDIRECTORY(TemplateExtOpenCV)
MACRO_OPTIONAL_ADD_SUBDIRECTORY(TemplateExt)
```

- set Path to [OpenCV](#)
- set Path to [vvvv](#)
- if it's true, your IDE (i.e VisualStudio) will copy your Plugin to [<vvvvPath>\freeframe](#)
- add subdirectories here (is done allready), this will be your Projects

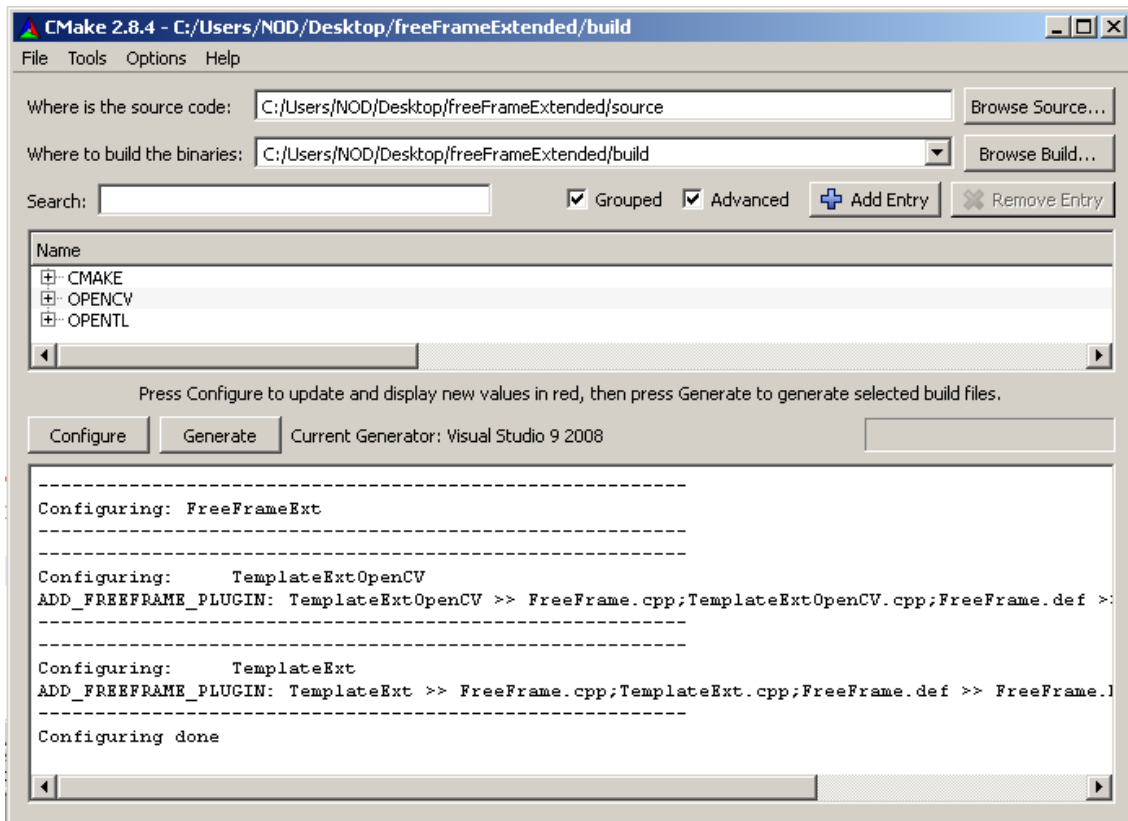
Make



- open CMake (cmake-gui)
- set source to [freeFrameExtended\source](#)
- set build to [freeFrameExtended\build](#)
- click configure



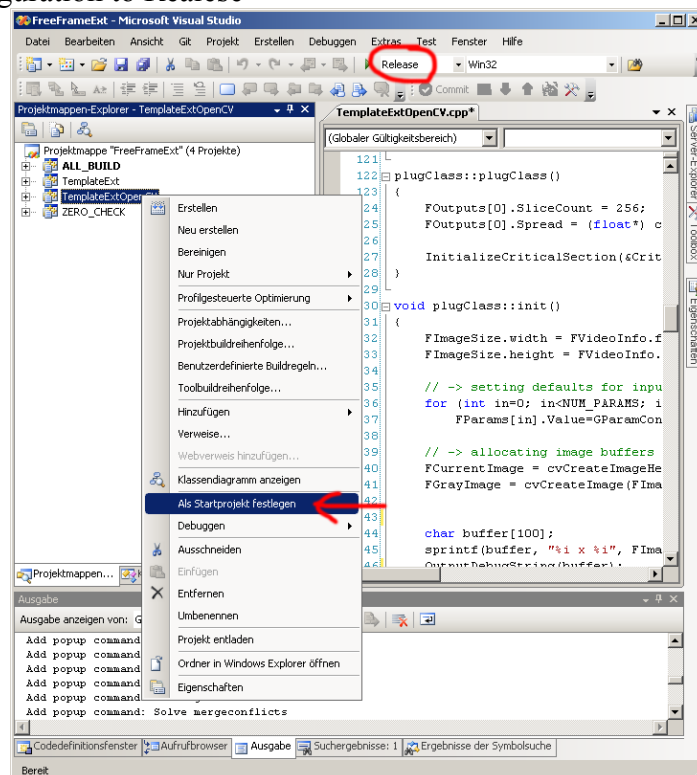
- select you IDE (i.e. VisualStudio) click [Finish](#)



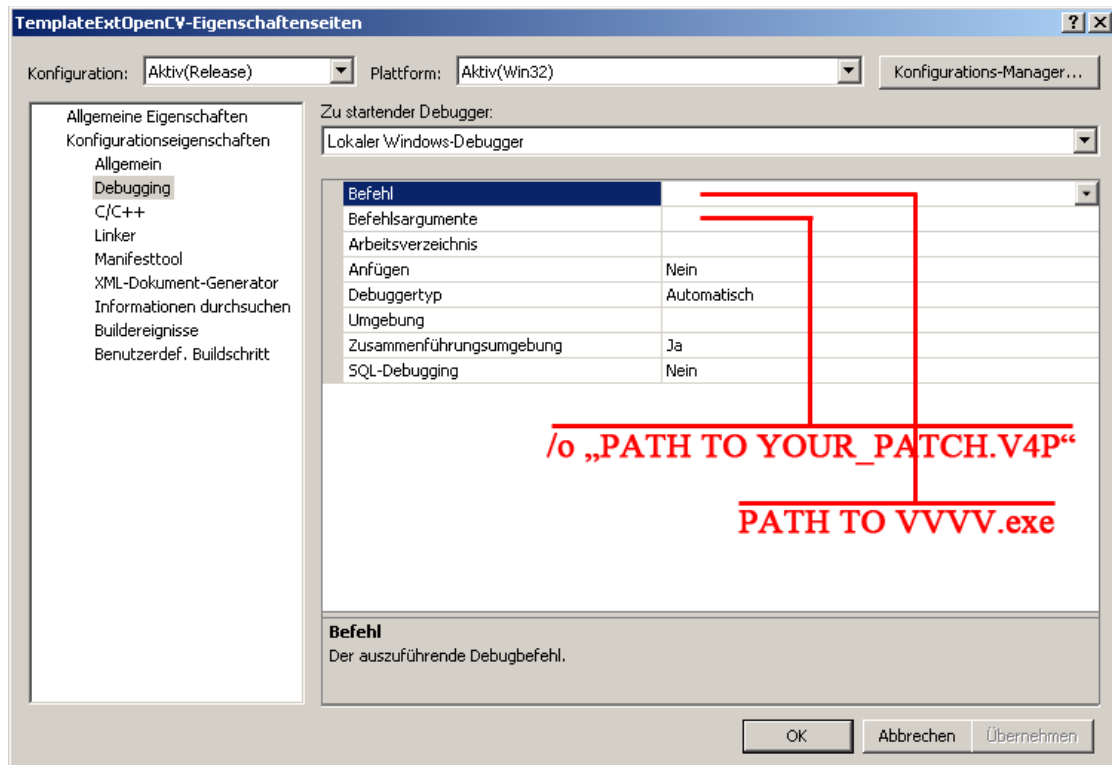
- Click [Generate](#)
- and your VisualStudioProject will be make
- you will find it at
- [freeFrameExtended/build\FreeFrameExt.sln](#)

Configure VisualStudio

- set your PluginProject to startProject
- set your Configuration to Release



- go to Project-Properties/Debugging/Comand
- set path to vvvv.exe
- set /o path to your Patch.v4p



- now you can RUN/DEBUG your PlugIn
- hit **F5** the PlugIn will be build to `freeFrameExtended\freeframe`)
- if you use `COPY_TO_VVVV` your patch will be open AND DEBUG
- IN CODE USE SOMTHING LIKE THAT TO DEBUG

```
char buffer[100];
sprintf(buffer, "%i x %i", FImageSize.width, FImageSize.height);
OutputDebugString(buffer);
```

Add CODE to your Project / Rename you Project

- if you want add a *.h or *.cpp Source File to your Project
- creat a txt File in your PluginProject (i.e freeFrameExtended\source\TemplateExtOpenCV)
- rename it to yourName.h or yourName.cpp
- open freeFrameExtended\source\TemplateExtOpenCV\CMakeLists.txt

```
# SET YOUR PLUGIN NAME
SET(MY_LIBRARY_NAME
    TemplateExtOpenCV                Rename Project Here
)

# ADD HERE YOUR SOURCE FILES
SET(SOURCE
    FreeFrame.cpp
    TemplateExtOpenCV.cpp
    FreeFrame.def                    add Source Files here
)

# ADD HERE YOUR HEADER FILES
SET(HEADERS
    FreeFrame.h
    TemplateExtOpenCV.h            add Header Files here
)

# USE OPENCV
SET(USE_OPENCV TRUE)              use OpenCV ??

# DONT CHANGE IT WILL CONFIGURE YOUR PROJECT
INCLUDE(MacroMyLibrary)
MACRO_MY_LIBRARY()

# COPY PLUGIN TO VVVV\freeframe
IF(COPY_TO_VVVV)
    ADD_CUSTOM_COMMAND(
        TARGET ${MY_LIBRARY_NAME}
        POST_BUILD
        COMMAND ${CMAKE_COMMAND} -E copy $
        {LIBRARY_OUTPUT_PATH}/Release/${MY_LIBRARY_NAME}.dll $
        {VVVV_PATH}/freeframe/${MY_LIBRARY_NAME}.dll
        COMMAND ${CMAKE_COMMAND} -E copy $
        {LIBRARY_OUTPUT_PATH}/Release/${MY_LIBRARY_NAME}.dll $
        {LIBRARY_OUTPUT_PATH}/${MY_LIBRARY_NAME}.dll
    )
ENDIF(COPY_TO_VVVV)
```

- Now go to Cmake Configure/Generate
- Your VisualStudio Project will be updated

Convert Code::Block Projects

- copy all SourceFiles (*.cpp; *.h) to [freeFrameExtended\source\yourPlugin](#)
- copy FreeFrame.def from
 - [freeFrameExtended\source\TemplateExtOpenCV\](#) to
 - [freeFrameExtended\source\yourPlugin](#)
- copy CMakeLists.txt from
 - [freeFrameExtended\source\TemplateExtOpenCV\](#) to
 - [freeFrameExtended\source\yourPlugin](#)
- edit CmakeLists.txt replace MY_LIBRARY_NAME, SOURCE FILES and HEADER FILES with your Sourcefiles
- open [FreeFrame.cpp](#) replace this

```
#ifdef WIN32
extern "C" __declspec(dllexport) __stdcall plugMainUnion plugMain(DWORD functionCode,
LPVOID pParam, LPVOID instanceID)
#elif LINUX
```

- with

```
#ifdef WIN32
    #ifdef _MSC_VER
        extern "C" __declspec(dllexport) plugMainUnion __stdcall plugMain(DWORD
functionCode, LPVOID pParam, LPVOID instanceID)
    #elif __GNUC__
        extern "C" __declspec(dllexport) __stdcall plugMainUnion plugMain(DWORD
functionCode, LPVOID pParam, LPVOID instanceID)
    #endif
#elif LINUX
```

- now will it work
 - maybe you must looking for some [namespaces](#)